STOCHASTIC OPTIMIZATION OVER PARALLEL QUEUES: CHANNEL-BLIND SCHEDULING, RESTLESS BANDIT, AND OPTIMAL DELAY

by

Chih-ping Li

A Dissertation Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(ELECTRICAL ENGINEERING)

August 2011

Dedication

To Mom and Dad,

and my beloved wife Ling-chih

Acknowledgments

Pursuing a Ph.D. at USC has been the most rewarding experience in my life. For the most of it, I thank my advisor Michael J. Neely. Apart from being a great role model over the years, Mike urges me to boldly pursue difficult research problems, and provides me with generous guidance and support to develop elegant solutions. I thank Mike for giving me the opportunity to struggle, nurturing me as an independent researcher, and showing me the joy of doing good research. I have learned about myself and research more than I expected when I first arrived at USC, and I am forever indebted to Mike for this.

I thank Bhaskar Krishnamachari and Qing Zhao for introducing their work on cognitive radio networks to me during a seminar at USC. Their work gives me a brand new perspective on doing stochastic optimization over Markovian channels and, more generally, over frame-based systems. Such a paradigm shift leads to a large part of this dissertation.

I appreciate Bhaskar Krishnamachari, Giuseppe Caire, Rahul Jain, and Amy R. Ward for taking their time serving on both my qual exam and defense exam committee, and I thank them for their valuable comments. I thank Gerrielyn Ramos, Anita Fung, Mayumi Thrasher, and Milly Montenegro for their generous administrative support that makes my life hassle-free at the Communication Sciences Institute.

Working at "the 5th floor" of EEB is memorable because of my friends and officemates Longbo Huang, Rahul Urgaonkar, Ozgun Bursalioglu, Paihan Huang, Wei-cherng Liao, and Hoon Huh. I also thank Wei-jen Hsu, Shiou-Hung Chen, Chihhan Chou, I-Hsiang

Lin, Chi-hsien Lin, May-chen Kuo, and Yenting Lin for their friendship all the way from National Taiwan University to USC and Los Angeles.

Getting a Ph.D. would not have been possible without the constant support and understanding from my family. My dad has always been encouraging me to pursue a higher degree, and my mom supports every decision I make. This dissertation is dedicated to them. I also thank my parents-in-law for their understanding and support over the past two years and the years to come.

Finally, I want to thank my wife and best friend Ling-chih, who has always been accompanying me even during our five years of long distance relationship. Thank you, Ling-chih, for loving me as who I am, and for being the most supportive wife in the world.

Table of Contents

Dedication ii

Acknowledgments iii

List of Tables ix

List of Figures x

Abstract xii

Chapter 1: Introduction to Single-Hop Network Control 1
- 1.1 Partially Observable Wireless Networks . . . 2
  - 1.1.1 Q1: Dynamic Channel Probing . . . 3
  - 1.1.2 Q2: Exploiting Channel Memory . . . 3
  - 1.1.3 Q3: Throughput Utility Maximization over Markovian Channels . 4
- 1.2 Multi-Class Queueing Systems . . . 4
  - 1.2.1 Q4: Delay-Optimal Control in a Multi-Class M/G/1 Queue with Adjustable Service Rates . . . 5
- 1.3 Methodology . . . 6
  - 1.3.1 Achievable Region Approach . . . 6
  - 1.3.2 Lyapunov Drift Theory . . . 9
- 1.4 Contributions . . . 10
- 1.5 Dissertation Outline . . . 12
- 1.6 Bibliographic Notes . . . 12

Chapter 2: Dynamic Wireless Channel Probing 13
- 2.1 Network Model . . . 14
- 2.2 Motivating Examples . . . 17
- 2.3 Optimal Power for Stability . . . 20
  - 2.3.1 Full and Blind Network Capacity Region . . . 23
- 2.4 Dynamic Channel Acquisition (DCA) Algorithm . . . 24
  - 2.4.1 Example of Server Allocation . . . 27
- 2.5 Simulations . . . 28
  - 2.5.1 Multi-Rate Channels . . . 28
  - 2.5.2 I.I.D. ON/OFF Channels . . . 30

2.6 Generalization . . . 34
2.6.1 Timing Overhead . . . 34
2.6.2 Partial Channel Probing . . . 36
2.7 Chapter Summary and Discussions . . . 38
2.8 Bibliographic Notes . . . 39
2.9 Proofs in Chapter 2 . . . 40
2.9.1 Proof of Lemma 2.4 . . . 40
2.9.2 Proof of Lemma 2.6 . . . 41
2.9.3 Proof of Theorem 2.7 . . . 42
2.9.4 Proof of Theorem 2.11 . . . 48

Chapter 3: Exploiting Wireless Channel Memory 51
3.1 Network Model . . . 52
3.2 Round Robin Policy RR(M) . . . 56
3.2.1 Throughput Analysis . . . 59
3.2.2 Example of Symmetric Channels . . . 61
3.2.3 Asymptotical Throughput Optimality . . . 62
3.3 Randomized Round Robin Policy RandRR . . . 63
3.3.1 Achievable Network Capacity: An Inner Bound . . . 66
3.3.2 Outer Capacity Bound . . . 67
3.3.3 Example of Symmetric Channels . . . 72
3.3.4 A Heuristically Tighter Inner Bound . . . 73
3.4 Tightness of Inner Capacity Bound: Symmetric Case . . . 74
3.4.1 Preliminary . . . 75
3.4.2 Analysis . . . 78
3.5 Queue-Dependent Round Robin Policy QRR . . . 79
3.6 Chapter Summary and Discussions . . . 83
3.7 Bibliographic Notes . . . 83
3.8 Proofs in Chapter 3 . . . 85
3.8.1 Proof of Lemma 3.2 . . . 85
3.8.2 Proof of Lemma 3.6 . . . 85
3.8.3 Proof of Theorem 3.8 . . . 86
3.8.4 Proof of Lemma 3.18 . . . 88
3.8.5 Proof of Lemma 3.10 . . . 90
3.8.6 Proof of Lemma 3.19 . . . 92
3.8.7 Proof of Lemma 3.15 . . . 94
3.8.8 Proof of Theorem 3.17 . . . 95
3.8.9 Proof of Lemma 3.20 . . . 102

Chapter 4: Throughput Utility Maximization over Markovian Channels 105
4.1 Network Model . . . 108
4.2 Randomized Round Robin Policy RandRR . . . 108
4.3 Network Utility Maximization . . . 110
4.3.1 The QRRNUM policy . . . 110
4.3.2 Lyapunov Drift Inequality . . . 112

4.3.3 Intuition 114
4.3.4 Construction of QRRNUM 115
4.4 Performance Analysis 117
4.5 Chapter Summary and Discussions 118
4.6 Bibliographic Notes 119
4.7 Proofs in Chapter 4 119
4.7.1 Proof of Lemma 4.3 119
4.7.2 Proof of Theorem 4.4 120

Chapter 5: Delay-Optimal Control in a Multi-Class M/G/1 Queue with Adjustable Service Rates 127
5.1 Queueing Model 129
5.1.1 Definition of Average Delay 130
5.2 Preliminaries 132
5.3 Achieving Delay Constraints 134
5.3.1 Delay Feasible Policy DelayFeas 136
5.3.2 Construction of DelayFeas 136
5.3.3 Performance of DelayFeas 138
5.4 Convex Delay Optimization 140
5.4.1 Delay Proportional Fairness 140
5.4.2 Delay Fairness Policy DelayFair 141
5.4.3 Construction of DelayFair 142
5.4.4 Intuition on DelayFair 144
5.4.5 Performance of DelayFair 145
5.5 Delay-Constrained Optimal Rate Control 147
5.5.1 Dynamic Rate Control Policy DynRate 149
5.5.2 Intuition on DynRate 150
5.5.3 Construction of DynRate 151
5.5.4 Performance of DynRate 152
5.6 Cost-Constrained Convex Delay Optimization 154
5.6.1 Cost-Constrained Delay Fairness Policy CostDelayFair 155
5.6.2 Construction of CostDelayFair 156
5.6.3 Performance of CostDelayFair 158
5.7 Simulations 158
5.7.1 DelayFeas and DelayFair Policy 158
5.7.2 DynRate and CostDelayFair Policy 160
5.8 Chapter Summary and Discussions 164
5.9 Bibliographic Notes 165
5.10 Additional Results in Chapter 5 167
5.10.1 Proof of Lemma 5.4 167
5.10.2 Lemma 5.19 168
5.10.3 Proof of Lemma 5.14 169
5.10.4 Independence of Second-Order Statistics in DynRate 171
5.10.5 Proof of Lemma 5.17 172
5.10.6 Proof of Theorem 5.18 173

Chapter 6: Conclusions 175

Bibliography 177

List of Tables

5.1 Simulation for the DelayFair policy under different values of V. 160
5.2 Simulation for the DynRate policy under different values of V. 162
5.3 Simulation for the CostDelayFair policy under different values of V. 163

List of Figures

2.1 The network capacity region Λ and the blind network capacity region Λ_{blind} in a two-user network. 20
2.2 Average power of DCA and optimal pure policies for $P_{\text{meas}} = 0$. The curves of purely channel-aware and DCA overlap each other. 29
2.3 Average power of DCA and optimal pure policies for $P_{\text{meas}} = 10$. 29
2.4 Average power of DCA and optimal pure policies for $P_{\text{meas}} = 5$. 30
2.5 Average power of DCA and optimal pure policies for different values of P_{meas}. The power curve of DCA overlaps with others at both ends of P_{meas} values. 31
2.6 Sample backlog processes in the last 10^5 slots of the DCA simulation. 32
2.7 Average backlogs of the three users under DCA. 33
2.8 Average power of DCA and optimal pure policies for different values of V. We set $\rho = 0.07$. 34
2.9 The average backlog of DCA under different values of V; we set $P_{\text{meas}} = 10$. 35
2.10 The new network capacity regions Λ_{new} of a two-user network with channel probing delay, Bernoulli ON/OFF channels, and a server allocation constraint. 35
3.1 A two-state Markov $\mathsf{ON/OFF}$ chain for channel $n \in \{1, 2, \dots, N\}$. 53
3.2 The k-step transition probabilities $\mathsf{P}_{n,01}^{(k)}$ and $\mathsf{P}_{n,11}^{(k)}$ of a positively correlated Markov $\mathsf{ON/OFF}$ channel. 56
3.3 The mapping f_n from information states $\omega_n(t)$ to modes $\{\mathsf{M1}, \mathsf{M2}\}$. 68
3.4 Mode transition diagrams for the real channel n. 69
3.5 Mode transition diagrams for the fictitious channel n. 69
3.6 Comparison of throughput regions under different assumptions. 73
3.7 Comparison of our inner capacity bound Λ_{int}, the unknown network capacity region Λ, and a heuristically better inner capacity bound $\Lambda_{\text{heuristic}}$. 74
3.8 An example for Lemma 3.15 in the two-user symmetric network. Point B and C achieve sum throughput $c_1 = \pi_{\mathsf{ON}} = 0.5$, and the sum throughput at D is $c_2 \approx 0.615$. Any other boundary point of Λ_{int} has sum throughput between c_1 and c_2. 77
5.1 Simulation for the $\mathsf{DelayFeas}$ policy under different delay constraints (d_1, d_2). 159
5.2 The performance region $\mathcal{W}$ of mean queueing delay in the simulations for the $\mathsf{DynRate}$ and the $\mathsf{CostDelayFair}$ policy. 161

5.3 The two dotted lines passing point C on AB represent two arbitrary randomized policies that achieve C. Geometrically, they have the same mixture of one point on $\mathcal{W}(25)$ and one on $\mathcal{W}(16)$. Therefore, they incur the same average service cost. 163

Abstract

This dissertation addresses several optimal stochastic scheduling problems that arise in partially observable wireless networks and multi-class queueing systems. They are single-hop network control problems under different channel connectivity assumptions and different scheduling constraints. Our goals are two-fold: To identify stochastic scheduling problems of practical interest, and to develop analytical tools that lead to efficient control algorithms with provably optimal performance.

In wireless networks, we study three sets of problems. First, we explore how the energy and timing overhead due to channel probing affects network performance. We develop a dynamic channel probing algorithm that is both throughput and energy optimal. The second problem is how to exploit time correlations of wireless channels to improve network throughput when instantaneous channel states are unavailable. Specifically, we study the network capacity region over a set of Markov ON/OFF channels with unknown current states. Recognizing that this is a difficult restless multi-armed bandit problem, we construct a non-trivial inner bound on the network capacity region by randomizing well-designed round robin policies. This inner bound is considered as an *operational* network capacity region because it is large and easily achievable. A queue-dependent round robin policy is constructed to support any throughput vector within the inner bound. In the third problem, we study throughput utility maximization over partially observable Markov ON/OFF channels (specifically, over the inner bound provided in the previous problem). It has applications in wireless networks with limited channel probing capability, cognitive radio networks, target tracking of unmanned aerial

vehicles, and restless multi-armed bandit systems. An admission control and channel scheduling policy is developed to achieve near-optimal throughput utility within the inner bound. Here we use a novel ratio MaxWeight policy that generalizes the existing MaxWeight-type policies from time-slotted networks to frame-based systems that have policy-dependent random frame sizes.

In multi-class queueing systems, we study how to optimize average service cost and per-class average queueing delay in a nonpreemptive multi-class $M/G/1$ queue that has adjustable service rates. Several convex delay penalty and service cost minimization problems with time-average constraints are investigated. We use the idea of virtual queues to transform these problems into a new set of queue stability problems, and the queue-stable policies are the desired solutions. The solutions are variants of dynamic $c\mu$ rules, and implement weighted priority policies in every busy period, where the weights are functions of past queueing delays in each job class.

Throughout these problems, our analysis and algorithm design uses and generalizes an achievable region approach driven by Lyapunov drift theory. We study the performance region (in throughput, power, or delay) of interest and identify, or design, a policy space so that every feasible performance vector is attained by a stationary randomization over the policy space. This investigation facilitates us to design queue-dependent network control policies that yield provably optimal performance. The resulting policies make greedy and dynamic decisions at every decision epoch, require limited or no statistical knowledge of the system, and can be viewed as learning algorithms over stochastic queueing networks. Their optimality is proved without the knowledge of the optimal performance.

Chapter 1

Introduction to Single-Hop Network Control

A single-hop network consists of a server relaying data to a set of users. Data destined for each user arrives randomly at the server, and is stored in a user-specific queue for eventual service. The server and each user are connected through a channel with random or constant connectivity. Channel states determine the maximum rate at which data can be reliably transferred across the channel. At each decision epoch, the server makes two sets of decisions. First, an admission control policy may be deployed so that only part of the arriving traffic is accepted. One reason of doing so is to prevent unsupported arrivals from overloading the network and creating unbounded queues. Second, the server schedules data transmissions to users until the next decision epoch. These decisions may depend on full or partial information of current channel states and current queueing information, and are possibly restricted by channel interference constraints, feasible power allocations, or predefined performance requirements such as mean delay or average power consumption. Overall, we study how to base control decisions on network states to optimize given performance objectives.

In this dissertation, we explore four stochastic scheduling problems using this single-hop model. The first three are in the context of scheduling over partially observable wireless channels (a single-hop network with random channel connectivity), and the last one is in the context of delay and rate-optimal priority scheduling over a multi-class $M/G/1$ queue with adjustable service rates (a single-hop network with constant channel connectivity).

1.1 Partially Observable Wireless Networks

To fully utilize a high-speed wireless network, we need to study its fundamental performance and develop efficient control algorithms. The key challenge is to accommodate time variations of channels (due to changing environments, multi-path fading, and mobility, etc.), and to adapt to other constraints such as channel interference or limited energy in mobile devices. It is now well known that *opportunistic scheduling*, which gives service priority to users with favorable channel conditions, can greatly improve network throughput. One example is the celebrated MaxWeight [Tassiulas and Ephremides 1993] throughput-optimal policy. This policy uses instantaneous channel state information and queueing information to schedule data transmissions, and is shown to support any achievable network throughput without the knowledge of traffic and channel statistics.

Previous work on opportunistic scheduling adopts some idealized assumptions. In a time-slotted wireless network, channel states are assumed i.i.d. over slots, and are instantaneously available by channel probing with negligible overheads. These assumptions, however, do not necessarily hold in practice. Furthermore, the same analytical techniques and MaxWeight policies do not necessarily work when these assumptions are removed. Specifically, while it is known that the MaxWeight opportunistic scheduling policy is capacity-achieving in ergodic but non-i.i.d. systems when channel states are always known in advance [Neely et al. 2005, Tassiulas 1997], the same is not true when channel states are unknown or only partially known. Therefore, we are motivated to study the challenging problems of control of wireless networks under more realistic assumptions. In particular, we look for practical performance-affecting factors previously overlooked, and explore how to adapt to them. Two such factors we address are *channel probing overhead* and *channel memory*. Specifically, we focus on the following three problems.

1.1.1 Q1: Dynamic Channel Probing

Channel state information, obtained by probing, plays a central role in modern high-performance wireless protocols. Practical channel probing, however, incurs overheads such as energy and delay. Energy is an importance resource in mobile devices and needs to be conserved. Delay due to probing degrades network throughput because the time used for probing cannot be re-used for data transmission. Therefore, it is important to understand how these overheads affect network performance, and how they can be optimized. Thus, we ask:

1. What is the set of all achievable throughput vectors, i.e., the network capacity region?

2. What is the minimum power, spent on channel probing and data transmission, to support any achievable throughput vector?

3. How to design intelligent channel probing and data transmission strategies to support any feasible throughput vector with minimum power consumption?

1.1.2 Q2: Exploiting Channel Memory

Following up on the previous problem involving partially observable wireless channels, we consider the next scenario. When instantaneous channel states are unavailable, it is wise to recognize that channel states may be time-correlated, i.e., have memory. Channel memory allows past channel observations to provide partial information of future states, and can potentially be used to improve performance. Thus we ask, when instantaneous current states are never known, what is the network capacity region assisted by channel memory? What are the throughput-optimal policies?

1.1.3 Q3: Throughput Utility Maximization over Markovian Channels

In the previous problem, we denote by Λ the network capacity region assisted by channel memory. Here, we consider solving throughput utility maximization problems over Λ. Specifically, we want to solve

$$\text{maximize: } g(\overline{\boldsymbol{y}}), \quad \text{subject to: } \overline{\boldsymbol{y}} \in \Lambda, \tag{1.1}$$

where $g(\cdot)$ represents a concave utility function and $\overline{\boldsymbol{y}}$ denotes a throughput vector.

The interest in throughput utility maximization over partially observable Markovian channels comes from its many applications, such as in: (1) limited channel probing in wireless networks; (2) improving spectrum efficiency in cognitive radio networks; (3) target tracking of unmanned aerial vehicles; (4) nonlinear optimization problems in restless multi-armed bandit systems. See Chapter 4 for more details.

1.2 Multi-Class Queueing Systems

In the fourth problem, we deviate from the above wireless control problems and consider a single-hop network with constant channels, i.e., a multi-class queueing system. Among its wide applications in manufacturing systems [Buzacott and Shanthikumar 1993], semiconductor factories [Shanthikumar et al. 2007], and computer communication networks [Kleinrock 1976], we are interested in control problems in modern computer systems and cloud computing environments. We are motivated by the scenarios below.

It is a common practice to have computers share processing power over different service flows. How to prioritize them to optimize a joint objective or provide differentiated services is an important problem to look at. Power dissipation is another relevant issue. Modern CPUs have increasingly dense transistor integration and can run at high clock rates. It results in high power dissipation that may jeopardize hardware stability and leads to massive electricity cost. The later problem is especially important in large-scale systems such as server farms or data centers [U.S. Environmental Protection Agency 2007]. We are thus concerned with jointly providing differentiated services and performing power-efficient scheduling in computers.

In cloud computing, a current trend is that companies outsource computing power to external providers such as Amazon (Elastic Compute Cloud; EC2), Google (App Engine), Microsoft (Azure), and IBM, and stop maintaining their internal computing facilities. This business model is mutually beneficial because service providers generate revenue by lending excessive processing power to companies, who in turn do not need to invest in local infrastructures that need regular maintenance and upgrade. From the viewpoint of the subscribing companies, it is then crucial to learn to optimally purchase computing power and distribute it intelligently to internal departments. Balancing the provision of differentiated services against average service cost is our focus.

1.2.1 Q4: Delay-Optimal Control in a Multi-Class M/G/1 Queue with Adjustable Service Rates

To model the above problems, we consider optimizing average service cost and per-class average queueing delay in a multi-class $M/G/1$ queue with adjustable service rates. Jobs that arrive randomly for service are categorized into N classes, and are served in a nonpreemptive fashion. We consider four delay-aware and rate allocation problems:

1. Satisfying an average queueing delay constraint in each job class; here we assume a constant service rate.

2. Minimizing a separable convex function of the average queueing delay vector subject to per-class delay constraints; assuming a constant service rate.

3. Under adjustable service rates, minimizing average service cost subject to per-class delay constraints.

4. Under adjustable service rates, minimizing a separable convex function of the average queueing delay vector subject to an average service cost constraint.

In computer applications, the first problem guarantees mean queueing delay in each job class. A motive for the second problem is to provide delay fairness across job classes. This is in the same spirit as the well-known rate proportional fairness [Kelly 1997] or utility proportional fairness [Wang et al. 2006], and we show in Section 5.4.1 that the objective function for delay proportional fairness is quadratic (rather than logarithmic). Since operating frequencies of modern CPUs are adjustable by dynamic power allocations [Kaxiras and Martonosi 2008], and power translates to electricity cost, the third problem seeks to minimize electricity cost subject to performance guarantees. From a different viewpoint, the fourth problem optimizes the system performance with a budget on electricity.

Cloud computing applications have similar stochastic optimization problems. The service cost there may be the price on computing power that service providers offer to subscribing companies.

1.3 Methodology

Our analytical method that solves the four wireless scheduling and queue control problems is a combination of achievable region approaches and *Lyapunov drift theory*.

1.3.1 Achievable Region Approach

An achievable region approach studies the set of all achievable performance in a control system (we only consider long-term average performance in this dissertation). It is a preliminary step of using optimization methods to design optimal control algorithms. A traditional wisdom, widely used in standard optimization theory, is to characterize a performance region by linear/nonlinear constraints on the performance measure. Here we take a new *policy-based* approach that facilitates the construction of network control algorithms. The key is to identify a collection of *base* policies (or decisions), so that every feasible performance vector can be attained by randomizing or time-sharing these base policies. In other words, there is a direct mapping from the performance region to the set of all randomizations of base policies. This way of characterizing a performance region is used in all problems we study.

Chapter 2: Dynamic Wireless Channel Probing

The performance measures in the channel probing problem are throughput and power. Here, we show that any feasible throughput vector can be supported with minimum power expenditure by a stationary randomized policy of channel probing and user scheduling that makes independently random decisions in every slot with a fixed probability distribution (see Theorem 2.7). Due to the i.i.d. nature of the network model, this result is proved by a sample-path analysis and the Law of Large Numbers.

Chapter 3 and 4: Scheduling over Partially Observable Markovian Channels

In Chapter 3 and 4, we are interested in the network capacity region Λ over a set of Markov ON/OFF channels with unknown current states. The main difficulty here is that the Markov ON/OFF channels do not seem to have enough structure for us to characterize Λ exactly. Another way of seeing it is that we try to characterize the set of all achievable time-average reward vectors in a restless multi-armed bandit system [Whittle 1988]. A direct approach to compute Λ is by locating all boundary points. Yet, this is difficult because each boundary point requires solving a high-dimensional Markov decision process [Bertsekas 2005] that suffers from the curse of dimensionality and a countably infinite state space (see details in Chapter 3). To further illustrate the difficulty of computing Λ, we consider a simple example of computing the boundary point in the direction $(1, 1, \dots, 1)$ over symmetric channels (i.e., the channels are independent and share the same transition probability matrix). Equivalently, we seek to maximize the network sum throughput over symmetric channels. Ahmad et al. [2009] shows that, in this case, the optimal policy is a greedy round robin policy that, on each channel, serves packets continuously until a failure occurs. The resulting sum throughput can be represented as

$$\mu_{\text{sum}} \triangleq \lim_{K\to\infty} \frac{\sum_{k=1}^{K}\sum_{n=1}^{N}(L_{kn}-1)}{\sum_{k=1}^{K}\sum_{n=1}^{N} L_{kn}},$$

where L_{kn} denotes the time interval in which channel n is served in the kth round (see Section 3.2.1 or Zhao et al. [2008]). Yet, in this simple example, the sum throughput μ_{sum} is still difficult to compute when $N > 2$. The reason is that the interval process $\{L_{11}, L_{12}, \ldots, L_{1N}, L_{21}, \ldots, L_{2N}, L_{31}, \ldots\}$ forms a high-order Markov chain.

Since computing the network capacity region Λ is difficult, an alternative we use in Chapter 3 and 4 is to construct an inner bound, denoted by Λ_{int}, on Λ. We want the inner capacity region Λ_{int} to have these nice properties: (1) The region is intuitively large. (2) It is easily achievable and scalable with the number of channels. (3) It has closed-form performance that is easy to analyze. (4) It is asymptotically optimal in some special cases. (5) Its structural properties provide insights on how channel memory improves throughput.

In Chapter 3, we construct an inner capacity region Λ_{int} that has all of the above properties. It is the achievable rate region of a well-designed class of *randomized round robin* policies (introduced in Section 3.3). These policies are motivated by the greedy round robin in Ahmad et al. [2009] that maximizes sum throughput over symmetric channels. With the above properties, we may consider Λ_{int} as an *operational* network capacity region, since achieving throughput vectors outside Λ_{int} may inevitably involve solving a much more complicated partially observable Markov decision process. Another attractive feature of constructing this inner bound Λ_{int} is that we can perform convex utility maximization over Λ_{int}. This is addressed in Chapter 4.

Chapter 5: Delay-Optimal Control in a Multi-Class M/G/1 Queue with Adjustable Service Rates

In a nonpreemptive multi-class $M/G/1$ queue with a constant service rate, the performance region of mean queueing delay is studied by Federgruen and Groenevelt [1988b], Gelenbe and Mitrani [1980]. In particular, Federgruen and Groenevelt [1988b] shows that the region is (a base of) a polymatroid [Welsh 1976], which is a special polytope with vertices achieved by strict priority policies. A useful consequence is that every feasible mean queueing delay vector is attainable by a stationary randomization of strict priority policies. Such randomization can be implemented frame by frame, where a permutation of $\{1, 2, \dots, N\}$ is randomly chosen in every busy period to prioritize job classes according to a fixed probability distribution. When the service rate of the $M/G/1$ queue is adjustable, we assume gated service allocations. That is, the service rate is fixed in every busy period, but may change across busy periods. The resulting performance region of mean queueing delay and average service cost corresponds to the set of all frame-based randomizations of strict priority policies with deterministic rate allocations.

1.3.2 Lyapunov Drift Theory

By associating a performance region with the set of all randomizations over a policy space, a network control problem is solved by an optimal time-sharing of these policies. We use Lyapunov drift theory to construct such a time-sharing that approximates the optimal solution and is an optimal policy.

Lyapunov drift theory was shown to stabilize general queueing networks in the landmark papers [Tassiulas and Ephremides 1992, 1993], and later used in Georgiadis et al. [2006], Neely [2003] to optimize general performance objectives such as average power [Neely 2006] or throughput utility [Eryilmaz and Srikant 2006, 2007, Georgiadis et al. 2006, Neely 2003, Neely et al. 2008, Stolyar 2005] in wireless networks. It is recently generalized to optimize frame-based systems that have policy-dependent random frame sizes [Li and Neely 2011b, Neely 2010c].

The control policy developed using Lyapunov drift theory has many useful features.

1. The sequence of base policies are created in a greedy and dynamic fashion, depending on the up-to-date network state information. In this regard, we may view it as a learning algorithm over stochastic queueing networks.

2. The policy requires limited statistical knowledge, such as arrival rates and channel statistics, of the network. In many applications, no statistical knowledge is needed.

3. The resulting policy has near-optimal performance, which can be proved without the knowledge of optimal performance.

1.4 Contributions

Chapter 2: Dynamic Wireless Channel Probing

Considering that channel probing takes time and energy, we characterize the new network capacity region. To support a feasible throughput vector, we show that a dynamic channel probing strategy is typically required, and we characterize the associated minimum power consumption. We propose a queue-dependent dynamic channel probing and rate allocation policy that supports any achievable throughput vector and stabilizes the network with minimum power consumption. Simulations are conducted to verify the performance.

Chapter 3: Exploiting Wireless Channel Memory

Knowing that the network capacity region over a set of partially observable Markovian channels is difficult to analyze, we construct a good inner capacity bound by randomizing well-designed round robin policies. The throughput gain from channel memory in this inner bound is discussed. In the case of symmetric channels, the tightness of the inner capacity bound is characterized by comparing it with an outer capacity bound, constructed using stochastic coupling methods and state aggregation to bound the performance of a restless multi-armed bandit problem using a related multi-armed bandit system. The bounds are tight when there are a large number of users and arrival traffic is symmetric. A queue-dependent dynamic round robin policy is designed to support every throughput vector within the inner capacity region and stabilize the network.

Chapter 4: Throughput Utility Maximization over Partially Observable Markovian Channels

Over the same network model in Chapter 3, we consider solving the throughput utility maximization problem (1.1). Since the full network capacity region Λ is difficult to analyze, we propose an approximation solution by solving the problem over the inner capacity bound Λ_{int} designed in Chapter 3. That is, we solve

$$\text{maximize:} \quad g(\overline{\boldsymbol{y}}), \quad \text{subject to:} \quad \overline{\boldsymbol{y}} \in \Lambda_{\text{int}}. \tag{1.2}$$

Equivalently, we provide an achievable-region method to optimize a nonlinear objective function and vector rewards over a restless multi-armed bandit system. We design a queue-dependent admission control and dynamic round robin policy that yields throughput utility that can be made arbitrarily close to the optimal solution of (1.2), at the expense of increasing average backlogs. The construction and analysis of this policy cannot be done using existing MaxWeight approaches. We propose a novel *ratio MaxWeight* policy to solve this problem. This new ratio rule generalizes existing MaxWeight methods from time-slotted networks to frame-based systems that have policy-dependent random frame sizes.

Chapter 5: Delay-Optimal Control in a Multi-Class M/G/1 Queue with Adjustable Service Rates

We provide a new queue-based method to solve the four delay-aware and rate control problems. This method uses the idea of virtual queues to transform the original problems into a new set of queue stability problems; queue-stable policies solve the original problems. The resulting policies are variants of dynamic $c\mu$ rules, and implement weighted priority policies in every busy period, where the weights are functions of past queueing delays in all job classes.

1.5 Dissertation Outline

Chapter 2 addresses the dynamic wireless channel probing problem. Chapter 3 explores the throughput gain by channel memory over partially observable Markovian channels. Chapter 4 proposes an achievable-region method to solve throughput utility maximization problems over partially observable Markovian channels. Chapter 5 addresses delay-optimal control problems in a nonpreemptive multi-class $M/G/1$ queue with adjustable service rates.

1.6 Bibliographic Notes

The results in Chapter 2 are published in Li and Neely [2007, 2010a]. The results in Chapter 3 are published in Li and Neely [2010b, 2011a]. The results in Chapter 4 are published in Li and Neely [2011b]. The results in Chapter 5 are under submission.

Chapter 2

Dynamic Wireless Channel Probing

In modern protocols over wireless networks, in which channels are random and time-varying, a common design is to persistently probe channels and use the channel state information to improve network performance such as throughput. An often overlooked issue is that channel probing in practice incurs a nonzero timing overhead. Since the time used for probing cannot be re-used for data transmission, throughput is practically degraded by probing delay. Therefore, channel probing creates a tradeoff in network throughput, and this tradeoff shall be optimized.

Channel probing also consumes energy from network devices such as power-constrained mobile phones. This is because probing requires information exchange between the server and users. In some cases, probing channels in every slot may be energy inefficient and undesired. For example, when the data arrival rate is low, transmitting data blindly (without channel state information) may suffice to support the rate. This requires a retransmission whenever a packet is sent over a bad channel. However, the energy incurred by retransmissions is, in some cases, less than the energy of constant channel probing.

The above observations motivate us to investigate dynamic channel probing strategies that are both throughput and energy optimal. This is the topic of this chapter. Throughput and energy optimal policies in wireless networks are previously studied in Neely [2006], which assumes instantaneous channel state information and no probing overheads. We will extend the analytical framework in Neely [2006] to incorporate dynamic channel probing. For ease of demonstration, in the first part of the chapter we assume that: (1) either all or none of the channels can be probed simultaneously; (2) probing delay is negligible. These assumptions are relaxed in Section 2.6.

The remainder of this chapter is organized as follows: Section 2.1 describes the network model. Motivating examples in Section 2.2 show the necessity of probing channels dynamically to optimize energy. The minimum average power to support any feasible throughput vector is characterized in Section 2.3. Our dynamic channel probing algorithm is developed in Section 2.4 and shown to be throughput and energy optimal. Simulation results are in Section 2.5. Section 2.6 presents generalized algorithms that consider channel probing delay and partial channel probing, where the assumption that all channels are probed is relaxed to allow any subset of the channels to be probed.

This chapter needs the following notations. For vectors $\boldsymbol{a} = (a_n)_{n=1}^N$ and $\boldsymbol{b} = (b_n)_{n=1}^N$,

- Let $\boldsymbol{a} \leq \boldsymbol{b}$ denote $a_n \leq b_n$ for all n; $\boldsymbol{a} \geq \boldsymbol{b}$ is defined similarly.
- Define the vector $\boldsymbol{a} \otimes \boldsymbol{b} \triangleq (a_1 b_1, \dots, a_N b_N)$.
- Let $\Pr[A]$ denote the probability of event A. Define the vector $\boldsymbol{Pr}(\boldsymbol{a} \leq \boldsymbol{b}) \triangleq (\Pr[a_1 \leq b_1], \dots, \Pr[a_N \leq b_N])$; $\boldsymbol{Pr}(\boldsymbol{a} \geq \boldsymbol{b})$ is defined similarly.
- Let $1_{[A]}$ be the indicator function of event A; $1_{[A]} = 1$ if A is true and 0 otherwise. Define the vector $\mathbf{1}_{[\boldsymbol{a} \leq \boldsymbol{b}]} \triangleq (1_{[a_1 \leq b_1]}, \dots, 1_{[a_N \leq b_N]})$.

2.1 Network Model

We consider a wireless base station serving N users through N time-varying channels. Time is slotted with normalized slots $t = [t, t+1)$, $t \in \mathbb{Z}^+$. Data is measured in integer units of packets. At the base station, we denote by $a_n(t)$ the number of packet arrivals destined for user $n \in \{1, \dots, N\}$ in slot t. We assume that $a_n(t)$ is i.i.d. over slots, independent across users, and independent of channel state processes. The term $a_n(t)$

takes values in $\{0, 1, 2, \ldots, A_{\max}\}$ with mean $\mathbb{E}[a_n(t)] = \lambda_n$ for all t, where $A_{\max}$ is a finite integer. Let $s_n(t)$ denote the channel state of user n in slot t. The state of every channel is fixed in every slot, and can only change at slot boundaries. We assume that $s_n(t)$ is i.i.d. over slots and takes integer values in $\mathcal{S} = \{0, 1, 2, \ldots, \mu_{\max}\}$, where $\mu_{\max} < \infty$. The value $s_n(t)$ represents the maximum number of packets that can be reliably transferred over channel n in slot t. Channel statistics are assumed known and fixed.

At the beginning of a slot, the base station either probes all channels with power expenditure P_{meas} (each channel measurement consumes P_{meas}/N units of power), or probes no channels (this assumption is relaxed in Section 2.6.2). We assume that channel probing is perfect and causes negligible delay (the later assumption is relaxed in Section 2.6.1). After possible probing, the base station allocates a service rate $\mu_n(t) \in \mathcal{S}$ to user $n \in \{1, \ldots, N\}$ in slot t. If $\mu_n(t) > 0$, a constant transmission power P_{tran} is consumed over channel n. The rate vector $\boldsymbol{\mu}(t) = (\mu_n(t))_{n=1}^N$ in every slot is chosen from a feasible set Ω, which includes the all-zero vector that corresponds to idling the system for one slot with no transmissions. The feasible set Ω may reflect channel interference constraints or other restrictions. For example, in a system that serves at most one user in every slot, every rate vector in Ω has at most one nonzero component.

When channel states $\boldsymbol{s}(t) = (s_n(t))_{n=1}^N$ are probed in slot t, a *channel-aware* rate vector $\boldsymbol{\mu}(t)$ is allocated according to $\boldsymbol{s}(t)$. Otherwise, a *channel-blind* rate vector $\boldsymbol{\mu}(t)$ is chosen oblivious of $\boldsymbol{s}(t)$. When $\mu_n(t) \leq s_n(t)$, indicating that the chosen service rate is supported by the current channel state, at most $\mu_n(t)$ packets can be delivered over channel n (limited by queue occupancy). Otherwise, we have $\mu_n(t) > s_n(t)$, and all transmissions over channel n fail in slot t. Such transmissions fail because the channel state leads to a signal-to-noise ratio that is insufficient to support the attempted transmission rate. For each feasible channel state vector $\boldsymbol{s}$, we define $\Omega(\boldsymbol{s}) \triangleq \{\boldsymbol{\mu} \in \Omega \mid \boldsymbol{\mu} \leq \boldsymbol{s}\}$. Without loss of generality, we assume that if the channel state vector $\boldsymbol{s}$ is probed in a slot, we allocate a channel-aware rate vector $\boldsymbol{\mu} \in \Omega(\boldsymbol{s})$ in that slot. At the end of each slot, an ACK/NACK feedback is sent from every served user to the base station over an independent error-free control channel. Absence of an ACK is regarded as a NACK. Such feedback is used for retransmission control.

In slot t, we define $\hat{\boldsymbol{\mu}}(t) \triangleq (\hat{\mu}_n(t))_{n=1}^N$ as the *effective* transmission rate vector, where for each n,

$$\hat{\mu}_n(t) \triangleq \begin{cases} \mu_n(t) = \mu_n(t, s_n(t)) & \text{if channel } n \text{ is probed,} \\ \mu_n(t)\, 1_{[\mu_n(t) \leq s_n(t)]} & \text{otherwise.} \end{cases} \tag{2.1}$$

We use the notation $\mu_n(t, s_n(t))$ (or $\boldsymbol{\mu}(t, \boldsymbol{s}(t))$ in vector form) to emphasize that $\boldsymbol{\mu}(t)$ is aware of and supported by $\boldsymbol{s}(t)$. The indicator function $1_{[\mu_n(t) \leq s_n(t)]}$ is required because of the possible blind transmission mode. Let $P_n(t)$ denote the sum of probing and transmission power consumed by user n in slot t. We have

$$P_n(t) = \begin{cases} P_{\text{meas}}/N + P_{\text{tran}}\, 1_{[\mu_n(t) > 0]} & \text{if channel } n \text{ is probed,} \\ P_{\text{tran}}\, 1_{[\mu_n(t) > 0]} & \text{otherwise.} \end{cases} \tag{2.2}$$

We suppose that a queue $Q_n(t)$ of infinite capacity is kept for each user n at the base station; it stores user-n data that arrived but yet to be served for eventual service. The value $Q_n(t)$ denotes the unfinished work of user n in slot t, and evolves across slots according to the queueing dynamics:

$$Q_n(t+1) = \max\left[Q_n(t) - \hat{\mu}_n(t), 0\right] + a_n(t), \tag{2.3}$$

where $\hat{\mu}_n(t)$ is defined in (2.1). Initially, we assume $Q_n(0) = 0$ for all n. We say the wireless downlink is (strongly) stable if

$$\limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{n=1}^{N} \mathbb{E}\left[Q_n(\tau)\right] < \infty.$$

Remark 2.1. While the above network model focuses on a wireless downlink, the same model can be directly applied to an uplink system, where a base station receives transmissions from N energy-aware wireless users. This can be seen as follows: A single queue is kept by each user to hold data to be transmitted (rather than keeping all N queues at the base station). On each slot, the base station coordinates by signaling different users to either measure or not measure their channels, and then allocates uplink rates to be used by the users. The goal for the base station is to support data rates within full capacity, while the sum power expenditure due to transmission and channel acquisition by the users is minimized.

2.2 Motivating Examples

First we give the following definitions:

Definition 2.2. A scheduling policy is called *purely channel-aware* if packets are scheduled for transmission in a slot only after channel states are acquired in that slot. A scheduling policy is called *purely channel-blind* if it never probes channels and serves packets without channel state information.[1]

Definition 2.3. Define the *blind network capacity region* Λ_{blind} as the closure of the set of all throughput vectors that can be supported by purely channel-blind policies. Define the *network capacity region* Λ as the closure of the set of all throughput vectors that can be supported by purely channel-aware policies.

Since purely channel-aware policies can emulate any policy that mixes channel-aware and channel-blind decisions, including purely channel-blind policies, the set Λ in Definition 2.3 is the *exact* network capacity region of the wireless downlink; i.e., it contains all throughput vectors that are supported by any scheduling policies. Notice that $\Lambda_{\text{blind}} \subset \Lambda$.

[1] Both purely channel-aware and purely channel-blind policies may take advantage of queue backlog information.

Next we compare the performance of purely channel-aware policies and that of purely channel-blind policies to motivate dynamic channel probing.

A Single-Queue Example

We consider a single queue served by a Bernoulli ON/OFF channel. In every slot, we have: (1) one packet arrives with probability λ and zero otherwise; (2) the channel is ON with probability q and OFF otherwise; (3) one packet can be served if the channel is ON. Over this queue, it is easy to see that purely channel-aware and purely channel-blind scheduling share the same network capacity region $\Lambda = \Lambda_{\text{blind}} = \{0 \leq \lambda \leq q\}$, but their power consumption to support a feasible data rate λ may differ. Purely channel-aware scheduling consumes average power $(P_{\text{meas}}/q) + P_{\text{tran}}$ to deliver a packet (it probes the channel $1/q$ times on average to see an ON state, and transmits a packet once). Thus, the average power to support λ is $\lambda(P_{\text{meas}}/q + P_{\text{tran}})$. On the other hand, purely channel-blind scheduling consumes average power P_{tran}/q to deliver a packet (it blindly transmits the same packet $1/q$ times on average for a successful transmission). Thus, it consumes average power $\lambda\, P_{\text{tran}}/q$ to support λ. To attain any data rate $\lambda \in (0, q)$, we prefer purely channel-aware scheduling than purely channel-blind scheduling if $(P_{\text{meas}}/q) + P_{\text{tran}} \leq P_{\text{tran}}/q$, i.e., $P_{\text{meas}}/P_{\text{tran}} \leq 1-q$, and we prefer purely channel-blind scheduling otherwise. This simple threshold rule indicates that purely channel-blind scheduling, depending on the power ratio and channel statistics, may outperform purely channel-aware scheduling.

A Multi-Queue Example

We consider allocating a server to N queues with independent Bernoulli ON/OFF channels. States of each channel are i.i.d. over slots. If a channel connects to the server and is turned ON, it can serve at most one packet. In our network model, it is equivalent to setting $\mu_{\max} = 1$ and $s_n(t) \in \{0, 1\}$ for all n and t. The feasible set Ω consists of all N-dimensional binary vectors that have at most one entry equal to 1. For each $n \in \{1, 2, \ldots, N\}$, define $q_n \triangleq \Pr[s_n(t) = 1]$ as the probability that channel n is ON. The following lemmas characterize the network capacity region Λ, the blind network capacity region Λ_{blind}, and the minimum power required to stabilize a data rate vector within Λ and Λ_{blind}.

Lemma 2.4 (Proof in Section 2.9.1)**.** The blind network capacity region Λ_{blind} is

$$\Lambda_{\text{blind}} = \left\{ \boldsymbol{\lambda} = (\lambda_n)_{n=1}^{N} \;\middle|\; \sum_{n=1}^{N} \frac{\lambda_n}{q_n} \leq 1 \right\}.$$

Over the class of purely channel-blind policies and for each arrival rate vector $\boldsymbol{\lambda}$ interior to Λ_{blind}, the minimum power to stabilize the network is equal to

$$\left(\sum_{n=1}^{N} \frac{\lambda_n}{q_n} \right) P_{\text{tran}}.$$

Lemma 2.5 (Theorem 1, Tassiulas and Ephremides [1993])**.** The network capacity region Λ consists of data rate vectors $\boldsymbol{\lambda} = (\lambda_n)_{n=1}^{N}$ satisfying, for every nonempty subset $\mathcal{N}$ of $\{1, 2, \ldots, N\}$,

$$\sum_{n \in \mathcal{N}} \lambda_n \leq 1 - \prod_{n \in \mathcal{N}} (1 - q_n).$$

Lemma 2.6 (Proof in Section 2.9.2)**.** Over the class of purely channel-aware policies and for each nonzero arrival rate vector $\boldsymbol{\lambda} = (\lambda_n)_{n=1}^{N}$ interior to Λ, the minimum power to stabilize the network is equal to

$$\left(\sum_{n=1}^{N} \lambda_n \right) P_{\text{tran}} + \theta^* P_{\text{meas}}, \quad \theta^* \triangleq \inf \left\{ \theta \in (0,1) \mid \boldsymbol{\lambda} \in \theta \Lambda \right\}.$$

Let us consider the special case $N = 2$ (see Fig. 2.1). Lemma 2.4 and 2.5 show that there is a capacity region difference between Λ and Λ_{blind}. The region Λ consists of all data rate vectors bounded by the outer thick solid line. The region Λ_{blind} consists of all data rate vectors bounded by the inner thick dotted line representing $\lambda_1/q_1 + \lambda_2/q_2 = 1$. Although data rate vectors within Λ_{blind} are supported by both purely channel-blind and purely channel-aware policies, the shaded areas in Fig. 2.1 illustrate areas in which purely

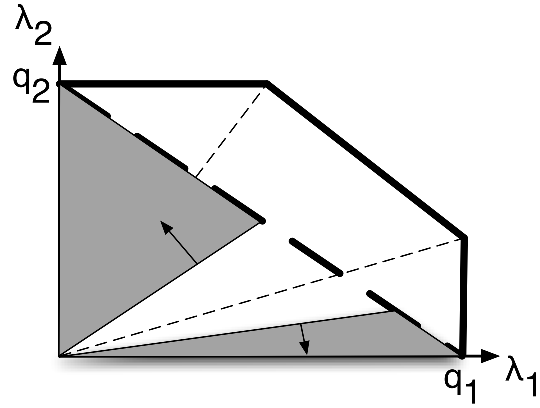

Figure 2.1: The network capacity region Λ and the blind network capacity region Λ_{blind} in a two-user network.

channel-blind scheduling is more energy efficient than purely channel-aware scheduling.[2] Interestingly, as the power ratio $P_{\text{meas}}/P_{\text{tran}}$ decreases (indicating that channel probing is getting cheaper), the shaded areas shrink in the specified directions. For arrival rate vectors outside Λ_{blind} but within Λ, purely channel-blind scheduling cannot stabilize the network, and we must use channel-aware transmissions.

In general, we show in the next section that an energy-optimal network-stable policy may be neither purely channel-blind nor purely channel-aware. Rather, mixed strategies with dynamic channel probing are typically required.

2.3 Optimal Power for Stability

For each arrival rate vector $\boldsymbol{\lambda}$ interior to the network capacity region Λ, we characterize the minimum power to stabilize $\boldsymbol{\lambda}$ when dynamic channel probing is allowed. We use sample-path arguments similar to the proof of Theorem 1 in Neely [2006]. In particular, we show that the minimum power to stabilize $\boldsymbol{\lambda}$ can be obtained by minimizing average power over a class of stationary randomized policies that yield time-average transmission

[2]The shaded areas are drawn under an additional assumption $q_1 \leq q_2$.

rates $\mu_n \geq \lambda_n$ for all users n. Each such stationary randomized policy makes decisions independently of queue backlog information, and has the following structure:

> In every slot, we probe all channels with some probability $\gamma \in [0, 1]$; otherwise no channels are probed. If a channel state vector $\boldsymbol{s}$ is probed, we randomly allocate a channel-aware transmission rate vector $\boldsymbol{\omega} \in \Omega(\boldsymbol{s})$ with some probability $\alpha(\boldsymbol{\omega}, \boldsymbol{s})$. If channels are not probed, we blindly allocate a rate vector $\boldsymbol{\omega} \in \Omega$ with some probability $\beta(\boldsymbol{\omega})$.

We recall that Ω denotes the set of all feasible rate allocation vectors, and $\Omega(\boldsymbol{s}) \subset \Omega$ denotes the subset of rate vectors supported by the channel state vector $\boldsymbol{s}$; $\Omega(\boldsymbol{s}) = \{\boldsymbol{\mu} \in \Omega \mid \boldsymbol{\mu} \leq \boldsymbol{s}\}$, where $\leq$ is taken entrywise.

Theorem 2.7 (Proof in Section 2.9.3)**.** For i.i.d. arrival and channel state processes with a data rate vector $\boldsymbol{\lambda}$ interior to Λ, the minimum power to stabilize the network is the optimal objective of the following optimization problem $\mathfrak{P}(\boldsymbol{\lambda})$ (defined in terms of auxiliary variables γ, $\alpha(\boldsymbol{\omega}, \boldsymbol{s})$ for each channel state vector $\boldsymbol{s} \in \mathcal{S}^N$ and each channel-aware rate vector $\boldsymbol{\omega} \triangleq (\omega_n)_{n=1}^N \in \Omega(\boldsymbol{s})$, and $\beta(\boldsymbol{\omega})$ for each feasible rate vector $\boldsymbol{\omega} \in \Omega$):

$$
\begin{aligned}
\text{minimize:}\quad & \gamma \sum_{\boldsymbol{s}\in\mathcal{S}^N} \pi_{\boldsymbol{s}} \left[\sum_{\boldsymbol{\omega}\in\Omega(\boldsymbol{s})} \alpha(\boldsymbol{\omega}, \boldsymbol{s}) \left(P_{\text{meas}} + \sum_{n=1}^{N} 1_{[\omega_n>0]} P_{\text{tran}} \right) \right] \\
& + (1-\gamma) \sum_{\boldsymbol{\omega}\in\Omega} \beta(\boldsymbol{\omega}) \left(\sum_{n=1}^{N} 1_{[\omega_n>0]} P_{\text{tran}} \right) && (\textit{average power}) \\
\text{subject to:}\quad & \boldsymbol{\lambda} \le \gamma \sum_{\boldsymbol{s}\in\mathcal{S}^N} \pi_{\boldsymbol{s}} \left(\sum_{\boldsymbol{\omega}\in\Omega(\boldsymbol{s})} \alpha(\boldsymbol{\omega}, \boldsymbol{s})\ \boldsymbol{\omega} \right) \\
& \qquad + (1-\gamma) \sum_{\boldsymbol{\omega}\in\Omega} \beta(\boldsymbol{\omega}) \Big(\boldsymbol{\omega} \otimes \boldsymbol{Pr}(\boldsymbol{S} \ge \boldsymbol{\omega}) \Big), && (\boldsymbol{\lambda} \le \boldsymbol{\mu}) \\
& \gamma \in [0,1] && (\textit{probing probability}) \\
& \beta(\boldsymbol{\omega}) \ge 0, \quad \forall\, \boldsymbol{\omega} \in \Omega && (\textit{blind tx probability}) \\
& \sum_{\boldsymbol{\omega}\in\Omega} \beta(\boldsymbol{\omega}) = 1 \\
& \alpha(\boldsymbol{\omega}, \boldsymbol{s}) \ge 0, \quad \forall\, \boldsymbol{\omega} \in \Omega(\boldsymbol{s}), \forall\, \boldsymbol{s} \in \mathcal{S}^N && (\textit{channel-aware tx probability}) \\
& \sum_{\boldsymbol{\omega}\in\Omega(\boldsymbol{s})} \alpha(\boldsymbol{\omega}, \boldsymbol{s}) = 1, \quad \forall \boldsymbol{s} \in \mathcal{S}^N
\end{aligned}
$$

where $\pi_{\boldsymbol{s}}$ denotes the steady state probability of channel state vector $\boldsymbol{s}$, and $\boldsymbol{S} \triangleq (S_n)_{n=1}^N$ denotes a random channel state vector. The vector notation $\boldsymbol{\omega} \otimes \boldsymbol{Pr}(\boldsymbol{S} \ge \boldsymbol{\omega})$ is defined before Section 2.1.

We denote by $P_{\text{opt}}(\boldsymbol{\lambda})$ the optimal objective of $\mathfrak{P}(\boldsymbol{\lambda})$; $P_{\text{opt}}(\boldsymbol{\lambda})$ is the minimum power to stabilize $\boldsymbol{\lambda}$. The next corollary is useful in the analysis of our dynamic channel probing policy proposed later.

Corollary 2.8. For i.i.d. arrival and channel state processes with an arrival rate vector $\boldsymbol{\lambda}$ interior to Λ, the stationary randomized policy that supports $\boldsymbol{\lambda}$ with minimum power allocates a service rate vector $\boldsymbol{\mu}(t)$ and consumes power $P_n(t)$ over channel $n \in \{1, \dots, N\}$ in every slot t such that

$$\sum_{n=1}^{N} \mathbb{E}\left[P_n(t)\right] = P_{\text{opt}}(\boldsymbol{\lambda}),\ \mathbb{E}\left[\hat{\boldsymbol{\mu}}(t)\right] \geq \boldsymbol{\lambda}, \quad \forall\, t \in \mathbb{Z}^+,$$

where $\hat{\boldsymbol{\mu}}(t)$ denotes the effective transmission rate vector, defined in (2.1).

2.3.1 Full and Blind Network Capacity Region

The network capacity region Λ and the blind network capacity region Λ_{blind} in Definition 2.3 can be characterized as corollaries of Theorem 2.7. Specifically, if we restrict the policy space to the class of purely channel-aware policies and neglect power, the proof of Theorem 2.7 characterizes Λ. Likewise, restricting the policy space to the class of purely channel-blind policies and neglecting power in Theorem 2.7 give us the blind region Λ_{blind}.

Corollary 2.9. For i.i.d. arrival and channel state processes, the network capacity region Λ of the wireless downlink consists of rate vectors $\boldsymbol{\lambda}$ satisfying the following. For each $\boldsymbol{\lambda}$, there exists a probability distribution $\{\alpha(\boldsymbol{\omega}, \boldsymbol{s})\}_{\boldsymbol{\omega} \in \Omega(\boldsymbol{s})}$ for each channel state vector $\boldsymbol{s}$ such that

$$\boldsymbol{\lambda} \leq \sum_{\boldsymbol{s} \in \mathcal{S}^N} \pi_{\boldsymbol{s}} \left(\sum_{\boldsymbol{\omega} \in \Omega(\boldsymbol{s})} \alpha(\boldsymbol{\omega}, \boldsymbol{s})\, \boldsymbol{\omega} \right),$$

where $\pi_{\boldsymbol{s}}$ is the steady state probability of channel state vector $\boldsymbol{s}$.

Corollary 2.9 is in the same spirit as [Neely et al. 2003, Theorem 1].

Corollary 2.10. For i.i.d. arrival and channel state processes, the blind network capacity region Λ_{blind} of the wireless downlink consists of rate vectors $\boldsymbol{\lambda}$ satisfying the following. For each $\boldsymbol{\lambda}$, there exists a probability distribution $\{\beta(\boldsymbol{\omega})\}_{\boldsymbol{\omega} \in \Omega}$ such that

$$\boldsymbol{\lambda} \leq \sum_{\boldsymbol{\omega} \in \Omega} \beta(\boldsymbol{\omega}) \Big(\boldsymbol{\omega} \otimes \boldsymbol{Pr}(\boldsymbol{S} \geq \boldsymbol{\omega}) \Big).$$

The blind capacity region in Lemma 2.4 is a special case of Corollary 2.10.

2.4 Dynamic Channel Acquisition (DCA) Algorithm

In Section 2.3, we established the minimum power required for network stability. Here, we develop a unified dynamic channel acquisition (DCA) algorithm that supports any throughput vector within the network capacity region Λ with average power that can be made arbitrarily close to optimal, at the expense of increasing average delay. This power-delay tradeoff is controlled by a predefined control parameter $V > 0$.

We define $\boldsymbol{\chi}(t) \triangleq [m(t), \boldsymbol{\mu}^{(c)}(t), \boldsymbol{\mu}^{(b)}(t)]$ as the collection of control variables in slot t. The variable $m(t)$ takes values in $\{0, 1\}$. All channels are probed in slot t if $m(t) = 1$; otherwise, no channels are probed. When channels are probed, channel-aware service rate vector $\boldsymbol{\mu}^{(c)}(t) = (\mu_n^{(c)})_{n=1}^N$ is allocated; otherwise, channel-blind rate vector $\boldsymbol{\mu}^{(b)}(t) = (\mu_n^{(b)})_{n=1}^N$ is allocated.

Dynamic Channel Acquisition (DCA) Algorithm:

In every slot t, we observe the current queue backlogs $\boldsymbol{Q}(t) = (Q_n(t))_{n=1}^N$ and choose the control $\boldsymbol{\chi}(t)$ that maximizes a function $f(\boldsymbol{Q}(t), \boldsymbol{\chi}(t))$ defined as

$$
\begin{aligned}
&f(\boldsymbol{Q}(t), \boldsymbol{\chi}(t)) \\
&\triangleq m(t)\Bigg\{ -V P_{\text{meas}} + \mathbb{E}_{\boldsymbol{s}}\left[\sum_{n=1}^{N} \Big(2\, Q_n(t)\, \mu_n^{(c)}(t) - V P_{\text{tran}}\, 1_{[\mu_n^{(c)}(t)>0]}\Big) \mid \boldsymbol{Q}(t)\right]\Bigg\} \\
&\quad + \overline{m}(t)\Bigg\{ \sum_{n=1}^{N} \Big[2\, Q_n(t)\, \mu_n^{(b)}(t) \Pr\Big[S_n \geq \mu_n^{(b)}(t)\Big] - V P_{\text{tran}}\, 1_{[\mu_n^{(b)}(t)>0]}\Big] \Bigg\},
\end{aligned}
\tag{2.4}
$$

where $\overline{m}(t) \triangleq 1 - m(t)$, $V > 0$ is a predefined control parameter, and the expectation $\mathbb{E}_{\boldsymbol{s}}[\cdot]$ is with respect to the randomness of the channel state vector $\boldsymbol{s}$.

The form of $f(\boldsymbol{Q}(t), \boldsymbol{\chi}(t))$ comes naturally from the underlying Lyapunov drift analysis. The design principle is that, to stabilize the network, we want to create a *negative drift* of queue backlogs whenever the backlogs are sufficiently large. Such negative drift keeps all queues bounded and stabilizes the network. We show later in Theorem 2.11 and its proof that maximizing (2.4) in every slot generates such a negative drift, balanced with average power consumption. The constant factor of 2 in (2.4) is a by-product of the analysis. It can be dropped by using a new constant $\widetilde{V} = V/2$ because maximizing (2.4) is equivalent to maximizing a positively scaled version of it.

Maximizing $f(\boldsymbol{Q}(t), \boldsymbol{\chi}(t))$ is achieved as follows. We separately maximize the multiplicands of $m(t)$ and $\overline{m}(t)$ in (2.4), and compare the maximum values. We choose $m(t) = 1$ if the optimal multiplicand of $m(t)$ is greater than that of $\overline{m}(t)$; otherwise, $m(t) = 0$. When choosing $m(t) = 1$, we acquire the current channel state vector $\boldsymbol{s}(t)$ and allocate a channel-aware rate vector $\boldsymbol{\mu}^{(c)}(t)$ that solves

$$\text{maximize:} \quad \sum_{n=1}^{N} \left(2\, Q_n(t)\, \mu_n^{(c)}(t) - V P_{\text{tran}}\, 1_{[\mu_n^{(c)}(t) > 0]} \right) \tag{2.5}$$

$$\text{subject to:} \quad \boldsymbol{\mu}^{(c)}(t) \in \Omega(\boldsymbol{s}(t)). \tag{2.6}$$

When $m(t) = 0$, we blindly allocate a rate vector $\boldsymbol{\mu}^{(b)}(t) \in \Omega$ that maximizes the multiplicand of $\overline{m}(t)$ in (2.4). The option of idling the system for one slot is included in taking $\overline{m}(t) = 1$ and $\boldsymbol{\mu}^{(b)}(t) = \mathbf{0}$.

Intuitively, the DCA algorithm estimates the expected gain of channel-aware and channel-blind transmissions by optimizing the multiplicands of $m(t)$ and $\overline{m}(t)$, and picks the one with a better gain in every slot. These estimations require joint and marginal distributions of channel states, and are the most complicated part of the algorithm. In (2.4), the multiplicand of $m(t)$ is a conditional expectation with respect to the randomness of channel states $\boldsymbol{s}(t)$. To maximize it, we need to solve (2.5)-(2.6) for each feasible channel state vector $\boldsymbol{s}(t)$, and then compute a weighted sum of the optimal solutions to (2.5)-(2.6) using the joint channel state distributions. Maximizing the multiplicand of $\overline{m}(t)$

is simpler and needs the marginal distribution of channel states. In practice, channel statistics may be estimated by taking samples of channel states. In Section 2.4.1, for the special case that at most one user is served in every slot, we show that the DCA algorithm can be performed in polynomial time, provided that channel statistics are known.

To analyze the performance of the DCA algorithm, it is useful to note that (2.4) can be re-written as

$$f(\boldsymbol{Q}(t), \boldsymbol{\chi}(t)) = \left(\sum_{n=1}^{N} 2\, Q_n(t)\, \mathbb{E}_{\boldsymbol{s}} \left[\hat{\mu}_n(t) \mid \boldsymbol{Q}(t) \right] \right) - V \mathbb{E}_{\boldsymbol{s}} \left[\sum_{n=1}^{N} P_n(t) \mid \boldsymbol{Q}(t) \right],$$

where $\hat{\mu}_n(t)$ and $P_n(t)$ are the effective service rate and the power consumption for user n in slot t, and

$$\hat{\mu}_n(t) \triangleq m(t)\, \mu_n^{(c)}(t) + \overline{m}(t)\, \mu_n^{(b)}(t)\, 1_{[\mu_n^{(b)}(t) \leq s_n(t)]},$$
$$P_n(t) \triangleq m(t) \left(\frac{P_{\text{meas}}}{N} + P_{\text{tran}}\, 1_{[\mu_n^{(c)}(t) > 0]} \right) + \overline{m}(t)\, P_{\text{tran}}\, 1_{[\mu_n^{(b)}(t) > 0]}.$$

Theorem 2.11. (Proof in Section 2.9.4) For any arrival rate vector $\boldsymbol{\lambda}$ interior to Λ and a predefined control parameter $V > 0$, the DCA algorithm stabilizes the network with average queue backlog and average power expenditure satisfying:

$$\limsup_{\tau \to \infty} \frac{1}{\tau} \sum_{t=0}^{\tau-1} \sum_{n=1}^{N} \mathbb{E}\left[Q_n(t) \right] \leq \frac{B + V(P_{\text{meas}} + N P_{\text{tran}})}{2\, \epsilon_{\max}}, \tag{2.7}$$

$$\limsup_{\tau \to \infty} \frac{1}{\tau} \sum_{t=0}^{\tau-1} \sum_{n=1}^{N} \mathbb{E}\left[P_n(t) \right] \leq \frac{B}{V} + P_{\text{opt}}(\boldsymbol{\lambda}), \tag{2.8}$$

where $B \triangleq (\mu_{\max}^2 + A_{\max}^2)N$ is a finite constant, $\epsilon_{\max} > 0$ is the largest value such that $(\boldsymbol{\lambda} + \boldsymbol{\epsilon}_{\max}) \in \Lambda$, where $\boldsymbol{\epsilon}_{\max}$ is an all-$\epsilon_{\max}$ vector.

The upper bounds in (2.7) and (2.8) are controlled by V. Choosing a sufficiently large V pushes the average power of the DCA algorithm arbitrarily close to the optimal $P_{\text{opt}}(\boldsymbol{\lambda})$, at the expense of linearly increasing the average congestion bound (or an average delay bound by Little's Theorem).

2.4.1 Example of Server Allocation

Consider an example of the N-queue downlink given in Section 2.2, where at most one queue serves at most one packet in every slot. The DCA algorithm, following the explanation in Section 2.4, can be simplified as follows. In each slot, if channels are probed, among all users with an ON channel state, we allocate the server to the user with the largest positive $f_n^{(c)}(t) \triangleq 2\,Q_n(t) - VP_{\text{tran}}$. If $f_n^{(c)}(t)$ is nonpositive for all users that have an ON state, we idle the server. Otherwise, channels are not probed, and we allocate the server to the user with the largest positive $f_n^{(b)}(t) \triangleq 2\,Q_n(t)\Pr[S_n = \text{ON}] - VP_{\text{tran}}$. If $f_n^{(b)}(t)$ is nonpositive for all users, we idle the server.

Next, to decide whether or not to probe channels, we compare the optimal multiplicands of $m(t)$ and $\overline{m}(t)$ in (2.4). Choosing the largest positive $f_n^{(b)}(t)$ yields the optimal multiplicand of $\overline{m}(t)$. The optimal multiplicand of $m(t)$ can be computed as

$$\begin{aligned} &-VP_{\text{meas}} + \sum_{n=1}^{N} (2\,Q_n(t) - VP_{\text{tran}})\,1_{[Q_n(t) > VP_{\text{tran}}/2]} \Pr\Big[s_n(t) = 1, \\ &\quad s_j(t) < \frac{Q_n(t)}{Q_j(t)},\, \forall j < n,\, s_k(t) \le \frac{Q_n(t)}{Q_k(t)},\, \forall k > n\Big]. \end{aligned} \tag{2.9}$$

This is is because we assign the server to user n when channel n is ON, $f_n^{(c)}(t) > 0$, and $Q_n(t)s_n(t) = Q_n(t) \ge s_j(t)Q_j(t)$ for all $j \neq n$ (we break ties by choosing the smallest index), which occurs with probability

$$\Pr\left[s_n(t) = 1, s_j(t) < \frac{Q_n(t)}{Q_j(t)},\, \forall j < n,\, s_k(t) \le \frac{Q_n(t)}{Q_k(t)},\, \forall k > n\right].$$

We probe channels if the optimal multiplicand of $m(t)$ is greater than that of $\overline{m}(t)$.

When channels are independent, we only need the marginal distribution of each channel to compute (2.9) in polynomial time. When channels have spatial correlations, (2.9) can also be easily computed in polynomial time, provided that the joint cumulative probability distribution is known or estimated.

2.5 Simulations

2.5.1 Multi-Rate Channels

We simulate the DCA algorithm in a server allocation problem in a symmetric three-user downlink defined as follows. Three users have independent Poisson arrivals with equal rates $\boldsymbol{\lambda} = \rho(1, 1, 1)$, where ρ is a scaling factor. Each user is served over an independent channel that is i.i.d. over slots and has three states $\{G, M, B\}$. In state G, M, and B, at most 2, 1 and 0 packets can be served, respectively. Define $q_G = 0.5$ as the probability that channel n is in state G in slot t. Probabilities $q_M = 0.3$ and $q_B = 0.2$ are defined similarly. In every slot, the controller picks at most one user to serve.

The maximum sum throughput of the downlink is

$$\begin{aligned}
&2 \cdot \Pr\left[\text{at least one channel is } G\right] \\
&+ 1 \cdot \Pr\left[\text{none of the channels is } G\text{, at least one is } M\right] \\
&= 2[1 - (1 - q_G)^3] + (1 - q_G)^3 - (1 - q_G - q_M)^3 = 1.867.
\end{aligned}$$

Thus, one face of the boundary of the network capacity region satisfies $\lambda_1 + \lambda_2 + \lambda_3 \leq 1.867$, which intersects with the scaled vector $\rho(1, 1, 1)$ at $\rho \approx 0.622$. In blind transmission mode, the maximum sum throughput of the downlink is equal to $2 \cdot q_G = 1$. As a result, one face of the boundary of Λ_{blind} is $\lambda_1 + \lambda_2 + \lambda_3 = 1$, which intersects $\rho(1, 1, 1)$ at $\rho \approx 0.33$. With this boundary information, we simulate the DCA algorithm for ρ from 0.05 to 0.6 with step size 0.05. Transmission power P_{tran} is 10 units, and each simulation is run for 10 million slots.

Fig. 2.2 and 2.3 compare the power consumption of DCA with the theoretical minimum power under purely channel-aware and purely channel-blind policies for different channel probing power $P_{\text{meas}} = \{0, 10\}$. V is set to 100. The theoretical minimum power of pure policies is computed by solving the optimization problem in Theorem 2.7. The power curve under purely channel-blind scheduling is drawn up to $\rho = 0.3$, a point close to the boundary of Λ_{blind}.

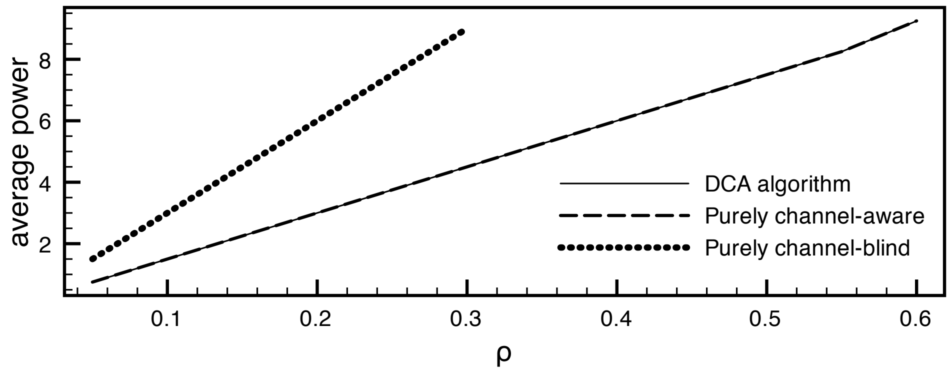

Figure 2.2: Average power of DCA and optimal pure policies for $P_{\text{meas}} = 0$. The curves of purely channel-aware and DCA overlap each other.

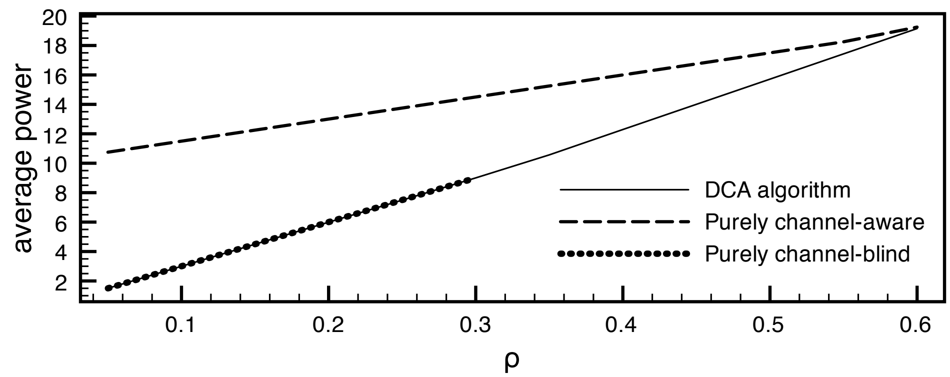

Figure 2.3: Average power of DCA and optimal pure policies for $P_{\text{meas}} = 10$.

When $P_{\text{meas}} = 0$, channel probing is cost-free, and it is always better to probe channels before allocating service rates. Therefore, purely channel-aware is no worse than any mixed strategies, and is optimal. Fig. 2.2 shows that the DCA algorithm consumes the same average power as the optimal purely channel-aware policy for all values of ρ. When $P_{\text{meas}} = 10$, the probing power is sufficiently large so that channel-blind transmissions are more energy efficient than channel-aware ones for $\boldsymbol{\lambda} \in \Lambda_{\text{blind}}$. In this case, Fig. 2.3 shows that the DCA algorithm performs as good as the optimal purely channel-blind policy when $\boldsymbol{\lambda} \in \Lambda_{\text{blind}}$ (i.e., $0 < \rho < 0.33$). When the arrival rates go beyond Λ_{blind} (i.e., $\rho \geq 0.33$), the DCA algorithm starts to incorporate channel-aware transmissions to stabilize the network, but still yields a significant power gain over purely channel-aware policies. These two cases show that, at extreme values of P_{meas}, the DCA algorithm is adaptive and energy optimal.

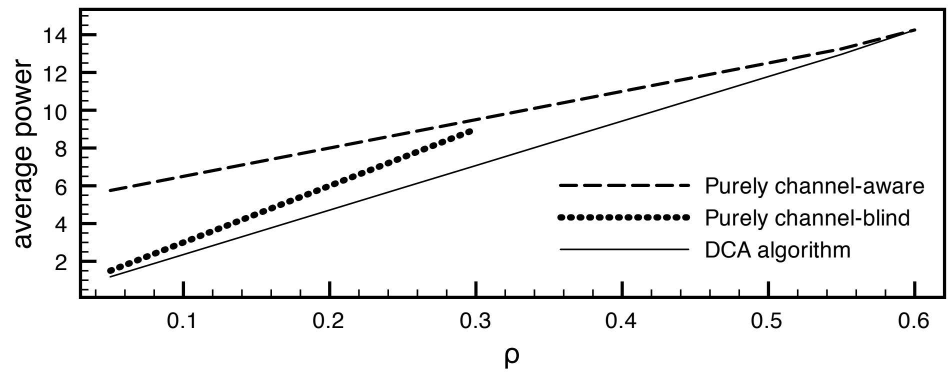

Figure 2.4: Average power of DCA and optimal pure policies for $P_{\text{meas}} = 5$.

Fig. 2.4 shows the performance of DCA for an intermediate power level $P_{\text{meas}} = 5$. In this case, the DCA algorithm outperforms both types of pure policies. To take a closer look, for a fixed arrival rate vector $\boldsymbol{\lambda} = (0.3, 0.3, 0.3)$, Fig. 2.5 shows the power gain of DCA over pure policies as a function of P_{meas} values. One important observation here is that DCA has the largest power gain when purely channel-aware and purely channel-blind have the same performance (around $P_{\text{meas}} = 4.5$). This is counter intuitive because, when the two types of pure policies perform the same, we expect mixing channel-aware and channel-blind decisions do not help. Yet, Fig. 2.5 shows that we benefit more from mixing strategies especially when one type of pure policy does not significantly outperform the other. In this example, DCA has as much as 30% power gain over purely channel-aware and purely channel-blind policies.

2.5.2 I.I.D. ON/OFF Channels

To have more insights on how DCA works, we perform another set of simulations. The setup is the same as the previous one except for the channel model. We suppose each

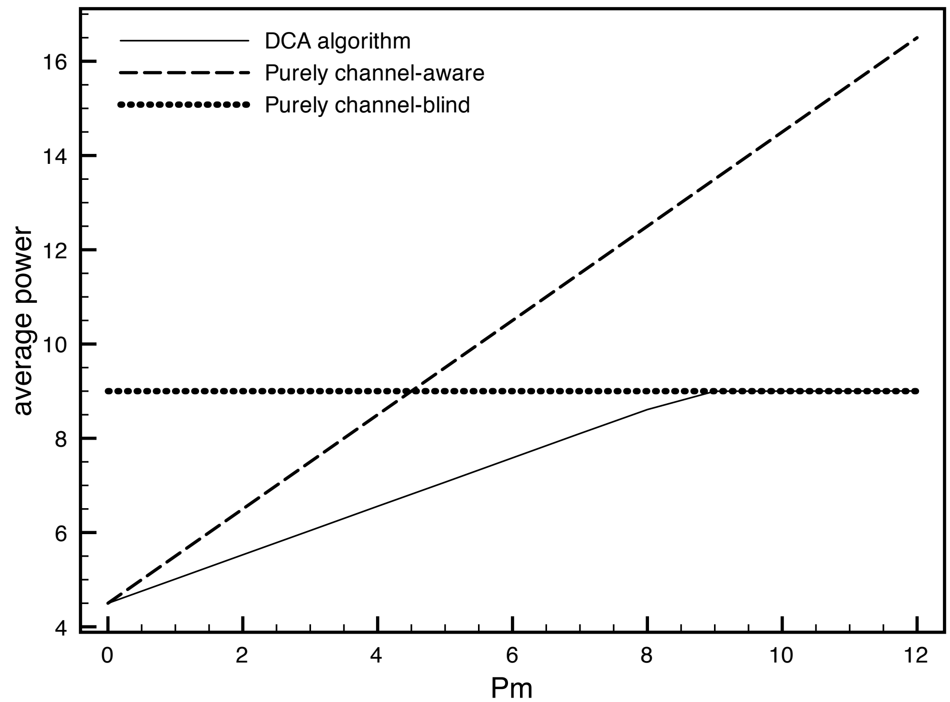

Figure 2.5: Average power of DCA and optimal pure policies for different values of P_{meas}. The power curve of DCA overlaps with others at both ends of P_{meas} values.

user is served by an independent i.i.d. Bernoulli ON/OFF channel. In every slot, channel 1, 2, and 3 is ON with probability 0.8, 0.5, and 0.2, respectively. When a channel is ON, one packet can be served over a slot; none is served when the channel is OFF. We simulate different arrival rate vectors of the form $\rho(3, 2, 1)$. It is easy to show that $\rho \approx 0.1533$ and 0.0784 correspond to the boundary of Λ and Λ_{blind}, respectively. We let $P_{\text{tran}} = 10$, and each simulation is run for 10 million slots.

User Backlogs

We first simulate the case $V = P_{\text{meas}} = 10$ and $\rho = 0.07$. Fig. 2.6 shows sample backlog processes of the three users in the last 10^5 slots of the simulation. We observe in (2.4) that we serve user 3 in channel-blind mode only if $U_3(t) \geq VP_{\text{tran}}/(2q_3) = 250$, and in channel-aware mode if $U_3(t) \geq VP_{\text{tran}}/2 = 50$. Fig. 2.6 shows that most of the time user 3 maintains its backlog between 50 and 250. It is consistent with our observation that user 3 operates mostly in channel-aware mode. In addition, the DCA algorithm generates a negative Lyapunov drift pushing the backlog of user 3 back under 250 whenever it is above 250. It explains why the user 3 backlog is kept around 250. Similar arguments can be made for the other two users, where the reference backlog level for starting a negative drift for user 1 and 2 are 62.5 and 100, respectively. We note that much of this backlog under reference levels can be eliminated by using a *place holder packet* technique in Neely and Urgaonkar [2008]. Indeed, from (2.4) we see that no packet is ever transmitted from queue n if $2Q_n(t) < VP_{\text{tran}}$, and so place holder packets reduce average backlog by roughly $VP_{\text{tran}}/2$ in each queue, without loss of energy optimality.

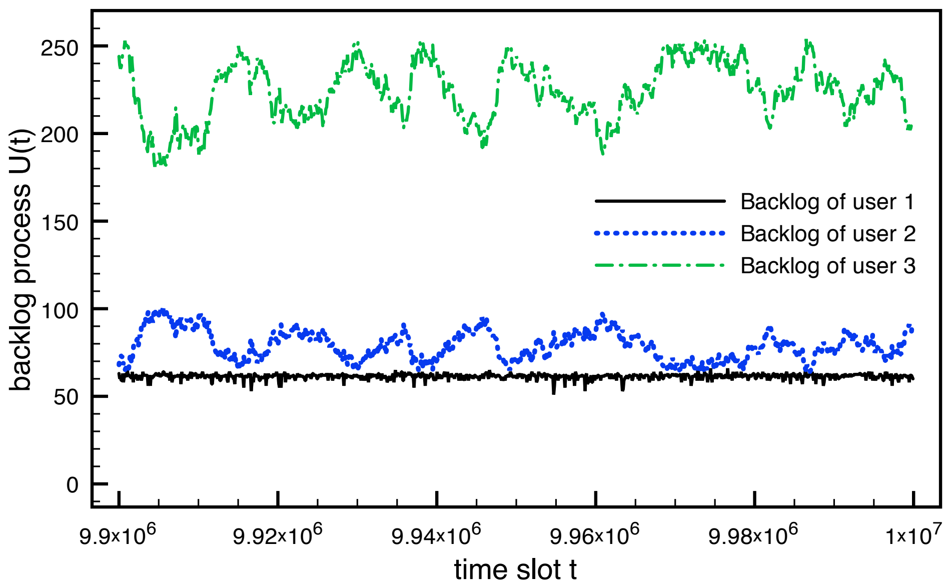

Figure 2.6: Sample backlog processes in the last 10^5 slots of the DCA simulation.

Next, for different values of ρ, Fig. 2.7 shows the average backlog of each user and the sum average backlog in the network. The DCA algorithm maintains roughly constant average backlogs (around the reference levels $VP_{\text{tran}}/(2\,q_n)$) for all users, except when data rates approach the boundary of the network capacity region; up to some point, the negative drift can no longer withhold the rapid increase of average backlog.

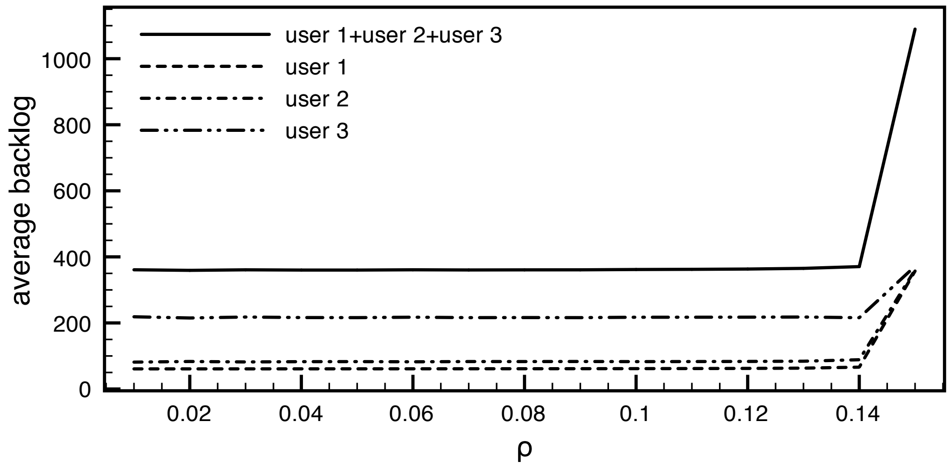

Figure 2.7: Average backlogs of the three users under DCA.

Control Parameter V

For different values of control parameter $V \in \{1, 10, 100\}$, Fig. 2.8 and 2.9 show the average power consumption of DCA (under different P_{meas} values) and the average sum backlogs. As V increases, the power consumption improves (but the improvement is diminishing) at the expense of increasing network delays. This result is consistent with (2.7) and (2.8).[3] Fig. 2.8 and 2.9 also show that, in practice, a moderate V value shall be chosen to maintain reasonable network delays without sacrificing much power consumption.[4]

---

[3]We note that performance bounds (2.7) and (2.8) are loose for Poisson arrivals because the associated $A_{\max} = \infty$ (and thus $B = \infty$). It is a minor concern because $A_{\max}$ is bounded with probability arbitrarily close to one, and the bound (2.7) is known to be loose.

[4]We note that, in Fig. 2.8, the DCA algorithm does not outperform purely channel-aware policies when $V = 1$ and P_{meas} close to zero. It may be because smaller V values do not allow DCA to maintain reasonable backlog levels in the queues. As a result, opportunistic scheduling gain cannot be fully exploited in channel-aware mode. It is a minor concern, but still suggests that a moderate V value is needed.

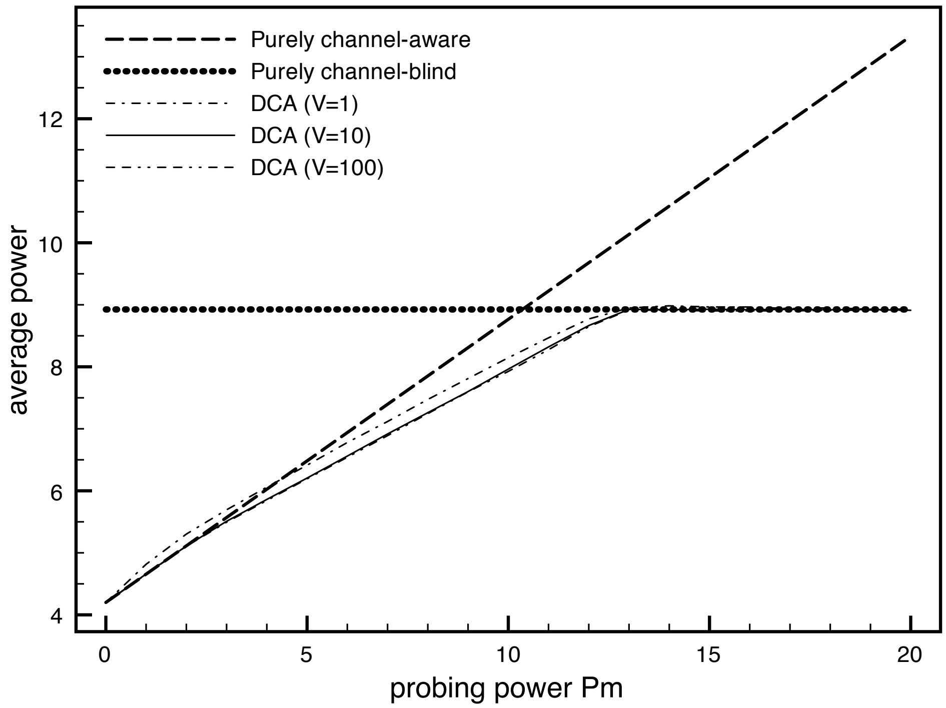

Figure 2.8: Average power of DCA and optimal pure policies for different values of V. We set $\rho = 0.07$.

2.6 Generalization

2.6.1 Timing Overhead

We generalize the network model in Section 2.1 to incorporate delay induced by channel probing. Specifically, we suppose that, in every slot, channel probing consumes $(1-\beta)$ of a slot for some $0 < \beta < 1$. If channels are probed and a service rate vector $\boldsymbol{\mu}(t)$ is allocated, the effective rate vector is $\beta\boldsymbol{\mu}(t)$. It follows that the class of purely channel-aware policies has a new scaled network capacity region $\beta\Lambda$, and it can no longer support the original Λ. In particular, there may be data rate vectors supported by purely channel-blind scheduling but not by purely channel-aware scheduling. The new network capacity region Λ_{new} under dynamic channel probing with timing overhead is the convex hull of

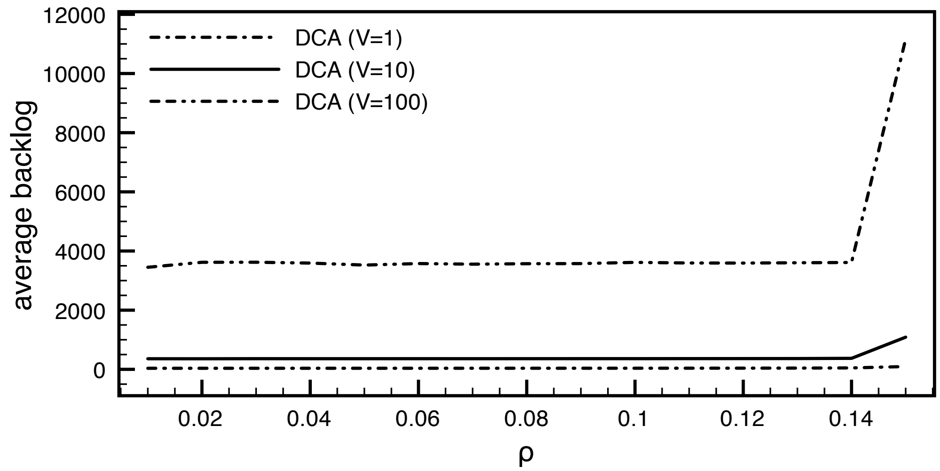

Figure 2.9: The average backlog of DCA under different values of V; we set $P_{\mathrm{meas}} = 10$.

the union of $\beta\Lambda$ and Λ_{blind}.[5] In other words, Λ_{new} consists of data rate vectors $\boldsymbol{\lambda}$ for each of which there exists a scalar $\gamma \in [0, 1]$ such that $\boldsymbol{\lambda} \in \gamma(\beta\Lambda) + (1-\gamma)\Lambda_{\mathrm{blind}}$.

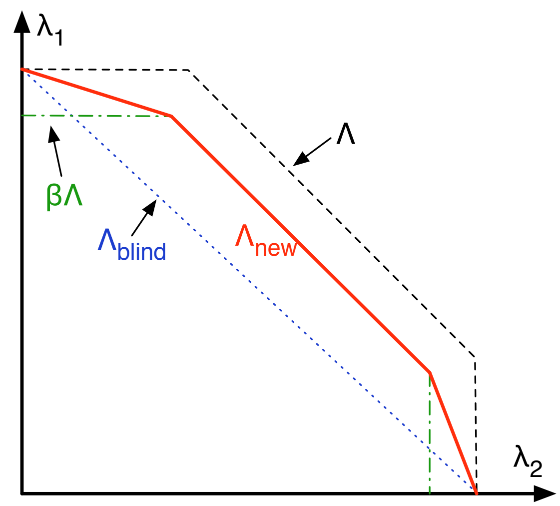

Figure 2.10: The new network capacity regions Λ_{new} of a two-user network with channel probing delay, Bernoulli ON/OFF channels, and a server allocation constraint.

As an example, we consider a two-user wireless downlink with the server allocation constraint described in Section 2.2. Its various network capacity regions are shown in

[5]It can be proved by following the proof of Theorem 2.7, where we neglect power and substitute every channel-aware rate vector $\boldsymbol{\mu}(t)$ with $\beta\boldsymbol{\mu}(t)$.

Fig. 2.10. Set Λ is the original network capacity region, scaled down to $\beta\Lambda$ when we consider channel probing delay. Set Λ_{blind} is the blind network capacity region, not affected by probing delay. Throughput vectors within $\Lambda_{\text{blind}} \setminus \beta\Lambda$ are supported by purely channel-blind policies, but not by purely channel-aware policies.

Generalizing the DCA algorithm to incorporate channel probing delay is easy. We only need to substitute every channel-aware rate allocation vector $\boldsymbol{\mu}(t)$ with $\beta\boldsymbol{\mu}(t)$. The new DCA algorithm applied to the above two-user downlink example is given below.

The DCA Algorithm under Probing Delay:

1. (Channel Probing) On each slot, define f_1 and f_2 as

$$\begin{aligned} f_1 &\triangleq -VP_{\text{meas}} + q_1\, q_2 \left[2\,\beta \max(Q_1(t), Q_2(t)) - VP_{\text{tran}}\right]^+ , \\ &\quad + q_1\, \overline{q}_2 \left[2\,\beta\, Q_1(t) - VP_{\text{tran}}\right]^+ + \overline{q}_1\, q_2 \left[2\,\beta\, Q_2(t) - VP_{\text{tran}}\right]^+ \\ f_2 &\triangleq \left[\max(2\, q_1\, Q_1(t), 2\, q_2\, Q_2(t)) - VP_{\text{tran}}\right]^+ , \end{aligned}$$

   where q_n denotes the probability that channel n is ON in a slot, $\overline{q_n} \triangleq 1 - q_n$, and $[x]^+ \triangleq \max(x, 0)$. We probe all channels if and only if $f_1 > f_2$.

2. (Server Allocation) When channels are probed, among those whose state is ON, serve the one with the largest positive $(2\,\beta\, Q_n(t) - VP_{\text{tran}})$. If all channels are OFF or $(2\,\beta\, Q_n(t) - VP_{\text{tran}}) \leq 0$ for all ON channels, serve no user for one slot. When channels are not probed, blindly serve the channel with the largest positive $(2\, q_n\, Q_n(t) - VP_{\text{tran}})$. Serve no user if $(2\, q_n\, Q_n(t) - VP_{\text{tran}}) \leq 0$ for all n.

2.6.2 Partial Channel Probing

Next, we relax the channel probing constraint and assume that any nonempty subset of channels can be probed in a slot. We denote by $J(t) \subset \{1, 2, \ldots, N\}$ the channel subset probed on slot t. For example, $J(t) = \{3, 5\}$ shows that the third and the fifth channel are probed on slot t. After partial channel probing, a service rate vector $\boldsymbol{\mu}(t) = (\mu_n(t))_{n=1}^N$ is allocated, such that $\mu_n(t)$ is channel-aware (i.e., $\mu_n(t) \le s_n(t)$) if $n \in J(t)$, and is chosen blindly otherwise. The effective service rate $\hat{\mu}_n(t)$ and the power expenditure $P_n(t)$ over channel n is written as

$$\hat{\mu}_n(t) = 1_{[n\in J(t)]}\mu_n(t) + 1_{[n\notin J(t)]}\mu_n(t)1_{[\mu_n(t)\le s_n(t)]}, \tag{2.10}$$

$$P_n(t) = 1_{[n\in J(t)]}\left(\frac{P_{\text{meas}}}{N} + P_{\text{tran}}1_{[\mu_n(t)>0]}\right) + 1_{[n\notin J(t)]}P_{\text{tran}}1_{[\mu_n(t)>0]}. \tag{2.11}$$

We can use the same Lyapunov drift analysis in the proof of Theorem 2.11 to modify the DCA algorithm to include partial channel probing. The main change is to substitute (2.10) and (2.11) into the drift analysis. The modified algorithm is given below.

Generalized DCA Algorithm:

1. (Channel Probing) In every slot t, for each nonempty subset $J(t)$ of $\{1,\dots,N\}$, we define $H_{J(t)} = \{(s_n(t))_{n\in J(t)} \mid s_n(t)\in\mathcal{S}\}$ as a collection of feasible channel state subvectors with respect to $J(t)$. We define $H_{J(t)} = \emptyset$ if $J(t)=\emptyset$. For each J(t) and each $h_{J(t)}\in H_{J(t)}$, we define $g^*(J(t),\boldsymbol{Q}(t),h_{J(t)})$ as the maximum of

$$\sum_{n\in J(t)}\left[2\,Q_n(t)\,\mu_n(t) - V\left(\frac{P_{\text{meas}}}{N} + P_{\text{tran}}1_{[\mu_n(t)>0]}\right)\right] \\ + \sum_{n\notin J(t)}\Big(2\,Q_n(t)\,\mu_n(t)\Pr\left[S_n \ge \mu_n(t) \mid h_{J(t)}\right] - VP_{\text{tran}}1_{[\mu_n(t)>0]}\Big). \tag{2.12}$$

subject to $(\mu_n(t))_{n=1}^N\in\Omega$ and $\mu_n(t)\le s_n(t)$ if $n\in J(t)$. Let $\boldsymbol{\mu}^*(J(t),\boldsymbol{Q}(t),h_{J(t)})$ be the maximizer of (2.12). Define

$$\underline{g}^*(J(t),\boldsymbol{Q}(t)) \triangleq \mathbb{E}\{g^*(J(t),\boldsymbol{Q}(t),h_{J(t)})\},$$

where the expectation is with respect to the randomness of $h_{J(t)}\in H_{J(t)}$. Define

$$\underline{g}^*(\boldsymbol{Q}(t)) \triangleq \max_{J(t)}\underline{g}^*(J(t),\boldsymbol{Q}(t)), \quad J^*(t) \triangleq \arg\max_{J(t)}\underline{g}^*(J(t),\boldsymbol{Q}(t)).$$

Then, if $\underline{g}^*(\boldsymbol{Q}(t)) > 0$, we probe channels in $J^*(t)$ in slot t; none is probed if $J^*(t) = \emptyset$. Otherwise, it is easy to show $\underline{g}^*(\boldsymbol{Q}(t)) = 0$, and we idle the system for a slot (including skipping Step 2 for one slot).

2. (Rate Allocation) In slot t, we denote by $h_{J^*(t)}$ the probed states over channels in $J^*(t)$. We allocate the service rate vector $\boldsymbol{\mu}^*(J^*(t), \boldsymbol{Q}(t), h_{J^*(t)})$ computed in Step 1.

In our original network model, feasible channel probing decision is binary: either probe all or none of the channels. Allowing partial channel probing expands the number of feasible decisions from 2 to 2^N, where N is the number of channels. It is not hard to show that, by generalizing Theorem 2.7, an energy-optimal policy to support a given throughput vector $\boldsymbol{\lambda}$ is a convex combination of 2^N stationary randomized policies, each of which corresponds to a channel subset $J(t)$. The generalized DCA algorithm evaluates a metric for each of the 2^N partial probing decisions, and uses one with the best metric in a slot. The resulting complexity of the generalized DCA algorithm is exponential. A compromise may be that we limit the decision space on partial channel probing to a predefined collection of channel subsets, with cardinality smaller than 2^N. This is still more general than our original network model, but the resulting DCA algorithm is of less complexity.

2.7 Chapter Summary and Discussions

Considering that channel probing consumes energy, we propose a Dynamic Channel Acquisition (DCA) algorithm that provides network stability with minimum energy consumption. This algorithm, developed via Lyapunov drift theory, is a unified treatment of incorporating both channel-aware and channel-blind transmissions to achieve throughput and energy optimality in wireless networks. Simulations show that the DCA algorithm adapts optimally to system parameters such as data arrival rates, transmission power, and channel probing power. Simulations also show that the DCA algorithm has the largest power gain when purely channel-aware and purely channel-blind strategies have the same performance, which is counter intuitive. We have discussed how to extend the DCA algorithm to incorporate nonzero timing overhead, which degrades network throughput. We have also discussed extensions that allow partial channel acquisition.

We show that the DCA algorithm is power and throughput optimal over channels with i.i.d. states over slots. However, it may not be optimal for general ergodic channels. Such channels have *memory*, in that past channel observations provide partial information of future states. An optimal policy shall exploit channel memory. The DCA algorithm does not base decisions on past channel observations, and therefore may not be optimal for general ergodic channels.

Despite the above, the DCA algorithm may still be a good suboptimal policy for general ergodic channels. It can be shown that the optimal power consumption $P_{\text{opt}}(\boldsymbol{\lambda})$ over channels without memory (see Theorem 2.7) can also be achieved in case of ergodic channels that have the same steady state distribution. Further, it can be shown that DCA achieves within $O(1/V)$ of this average power in the general ergodic case. While $P_{\text{opt}}(\boldsymbol{\lambda})$ is no longer optimal for channels with memory, the DCA algorithm can be viewed as optimizing over the restricted class of policies that neglect past channel history. Such restrictions simplify algorithm design when channel memory is complex to track.

In the next chapter, we show how to exploit channel memory to improve network throughput in a partially observable wireless network.

2.8 Bibliographic Notes

In the literature, Giaccone et al. [2003], Lee et al. [2006], Neely [2006], Neely et al. [2003], Yeh and Cohen [2004] focus on throughput/utility maximization with energy constraints in wireless networks, assuming that channel states are always known with negligible cost. Ji et al. [2004] and Sabharwal et al. [2007] show that measuring all channels regularly may not be throughput optimal because of the tradeoff between multi-user diversity gain and the associated timing overhead of channel probing. Kar et al. [2008] studies throughput-achieving algorithms when channel states are only measured every $T > 1$ slots. Gopalan et al. [2007] develops a MaxWeight-type throughput-optimal policy in a wireless downlink, assuming that only a subset of channels, chosen from a fixed collection of subsets, can be observed at any time and only the channels with known states can serve packets. Gesbert and Alouini [2004], Patil and de Veciana [2007], Tang and Heath [2005] study the performance of a wireless downlink for which a channel state is only sent from a user to the base station when the associated channel quality exceeds some threshold. Chang and Liu [2007], Guha et al. [2006a,b] develop optimal/near-optimal policies for joint partial channel probing and rate allocations to optimize a linear network utility.

2.9 Proofs in Chapter 2

2.9.1 Proof of Lemma 2.4

Consider a data rate vector $\boldsymbol{\lambda} = (\lambda_n)_{n=1}^N$ that can be stabilized by some purely channel-blind policy. A successful blind packet transmission over channel $n \in \{1, 2, \dots, N\}$ takes on average $1/q_n$ attempts. By Little's Theorem, the fraction of time the server is busy is equal to $\sum_{n=1}^N \lambda_n/q_n$, which must be less than or equal to 1 for stability. The *necessary* average power to support $\boldsymbol{\lambda}$ is equal to $(\sum_{n=1}^N \lambda_n/q_n)P_{\text{tran}}$.

Conversely, for each nonzero rate vector $\boldsymbol{\lambda}$ satisfying $\sum_{n=1}^N \lambda_n/q_n < 1$, we define $\rho \triangleq \sum_{n=1}^N \lambda_n/q_n$; there exists some $\epsilon > 0$ such that $\rho + \epsilon < 1$. Consider the policy that, in every slot, idles the system with probability $(1 - \rho - \epsilon)$ or servers user $n \in \{1, \dots, N\}$ with probability

$$\left(\frac{\rho + \epsilon}{\rho}\right)\left(\frac{\lambda_n}{q_n}\right).$$

This policy yields a service rate $(1 + \epsilon/\rho)\lambda_n > \lambda_n$ for each user n, and therefore stabilizes the system. The average power consumption of this policy is equal to $(\rho + \epsilon)P_{\text{tran}}$. By passing $\epsilon \to 0$, the rate vector $\boldsymbol{\lambda}$ can stabilized with average power arbitrarily close to $\rho P_{\text{tran}} = (\sum_{n=1}^{N} \lambda_n / q_n) P_{\text{tran}}$. It completes the proof. $\square$

2.9.2 Proof of Lemma 2.6

Suppose the rate vector $\boldsymbol{\lambda}$ can be stabilized by a purely channel-aware policy Φ. For simplicity, we assume Φ is ergodic with well-defined time averages (the general case can be proven similarly, as in Neely [2006]). Define θ as the fraction of time Φ probes the channels. The average power consumption of Φ is $(\sum_{n=1}^{N} \lambda_n) P_{\text{tran}} + \theta P_{\text{meas}}$. Suppose θ satisfies $0 < \theta < \theta^*$. A necessary condition to stabilize $\boldsymbol{\lambda}$ is that, for each nonempty subset $\mathcal{N} \subset \{1, 2, \dots, N\}$ of channels, the partial sum $\sum_{n \in \mathcal{N}} \lambda_n$ must be less than or equal to the fraction of time at least one channel in $\mathcal{N}$ can serve packets. In other words, for each $\mathcal{N}$, we have

$$\sum_{n \in \mathcal{N}} \lambda_n \leq \theta (1 - \prod_{n \in \mathcal{N}} (1 - q_n)).$$

In other words, $\boldsymbol{\lambda} \in \theta \Lambda$. But this contradicts the definition of θ^*, finishing the proof of necessity.

Conversely, since $\boldsymbol{\lambda} \in \theta^* \Lambda$, $\boldsymbol{\lambda}$ is an interior point of the set $(\theta^* + \epsilon)\Lambda$ for some $\epsilon > 0$ satisfying $\theta^* + \epsilon < 1$. We define a policy Φ as follows: On every slot, Φ probes the channels with probability $(\theta^* + \epsilon)$. When channels are probed in a slot, we serve the longest ON queue. Otherwise, we idle the system. It follows that applying policy Φ to the original wireless downlink is equivalent to applying the Longest Connected Queue (LCQ) policy [Tassiulas and Ephremides 1993] to a new wireless downlink in which a channel is ON if and only if it is probed and known to be ON. The capacity region of the new system is $(\theta^* + \epsilon)\Lambda$. Although channels in the new system are correlated (they are all OFF when probing is not performed), it is known that the LCQ policy is still throughput optimal over the new system [Tassiulas and Ephremides 1993]. As a result, if $\boldsymbol{\lambda}$ is interior to $(\theta^* + \epsilon)\Lambda$, policy Φ stabilizes $\boldsymbol{\lambda}$ with average power consumption equal to $(\sum_{n=1}^{N} \lambda_n)P_{\text{tran}} + (\theta^* + \epsilon)P_{\text{meas}}$. Passing $\epsilon \to 0$ finishes the sufficiency part of the proof. $\square$

2.9.3 Proof of Theorem 2.7

Suppose $\boldsymbol{\lambda}$ is interior to Λ. We first show the necessity part of the proof saying that there is no policy able to stabilize $\boldsymbol{\lambda}$ with average power strictly less than $P_{\text{opt}}(\boldsymbol{\lambda})$. Then we finish the sufficiency part of the proof by showing that the rate vector $\boldsymbol{\lambda}$ can be stabilized with average power arbitrarily close to $P_{\text{opt}}(\boldsymbol{\lambda})$.

(Necessity) Suppose the rate vector $\boldsymbol{\lambda}$ is stabilized by a policy Φ. This policy decides in which slots channels are probed, and allocates a transmission rate vector $\boldsymbol{\mu}(t)$ with power consumption $(P_n(t))_{n=1}^{N}$ in every slot t. In an interval $[0, M)$ for some integer M, we define

$$\hat{\boldsymbol{\mu}}_{\text{av}}(M) \triangleq \frac{1}{M} \sum_{\tau=0}^{M-1} \hat{\boldsymbol{\mu}}(\tau) \tag{2.13}$$

as the empirical service rate vector of policy Φ, where $\hat{\boldsymbol{\mu}}(\tau)$ is defined in (2.1). We define

$$P_{\text{av}}(M) \triangleq \frac{1}{M} \sum_{\tau=0}^{M-1} \sum_{n=1}^{N} P_n(\tau) \tag{2.14}$$

as the empirical average power of Φ. Define

$$\overline{P}_{\text{av}} \triangleq \liminf_{M\to\infty} P_{\text{av}}(M) \tag{2.15}$$

as a lower bound on the average power of Φ.[6] The value of $P_{\text{av}}(M)$ is bounded for all M because it is nonnegative and at most $(P_{\text{meas}} + NP_{\text{tran}})$. It follows that $\{P_{\text{av}}(M)\}_{M\in\mathbb{N}}$ is a bounded sequence and has finite limit points by Weierstrass's Theorem [Rudin 1976,

[6]When Φ is ergodic, limiting time averages are well-defined, and $\overline{P}_{\text{av}}$ is the exact limiting average power of Φ. If Φ is non-ergodic, limiting time averages may not exist, and $\overline{P}_{\text{av}}$ is a lower bound on the average power of Φ.

Theorem 2.42]. In addition, $\overline{P}_{\text{av}}$ is a limit point of $\{P_{\text{av}}(M)\}_{M\in\mathbb{N}}$ [Rudin 1976, Theorem 3.17], and there exists an integer subsequence $\{M_n\}_n$ such that

$$\lim_{n\to\infty} P_{\text{av}}(M_n) = \liminf_{M\to\infty} P_{\text{av}}(M) = \overline{P}_{\text{av}}. \tag{2.16}$$

Next, we show that, for any $\boldsymbol{\lambda}$-stabilizing policy Φ (ergodic or non-ergodic), there exists a stationary randomized policy $\hat{\Phi}$ that consumes average power $\overline{P}_{\text{av}}$ and yields an average service rate vector $\boldsymbol{\mu} \geq \boldsymbol{\lambda}$. The parameters of policy $\hat{\Phi}$ constitute a feasible solution to the optimization problem $\mathfrak{P}(\boldsymbol{\lambda})$. Thus, its average power $\overline{P}_{\text{av}}$, as well as the average power of the original $\boldsymbol{\lambda}$-stabilizing policy Φ, must be greater than or equal to the optimal objective $P_{\text{opt}}(\boldsymbol{\lambda})$ of $\mathfrak{P}(\boldsymbol{\lambda})$. Since this result holds for any $\boldsymbol{\lambda}$-stabilizing policy, the necessary power to stabilize $\boldsymbol{\lambda}$ is at least $P_{\text{opt}}(\boldsymbol{\lambda})$.

Let $T_M^{(c)}$ and $T_M^{(b)}$ be a subset of slots in $[0, M)$ in which channels are probed and not probed, respectively; we have $|T_M^{(c)}| + |T_M^{(b)}| = M$ for all M. Without loss of generality, we assume $T_M^{(c)}$ and $T_M^{(b)}$ are nonempty. According to (2.1), we define

$$\hat{\boldsymbol{\mu}}_{\text{av}}^{(c)}(M) \triangleq \frac{1}{M} \sum_{\tau\in T_M^{(c)}} \hat{\boldsymbol{\mu}}(\tau) = \frac{1}{M} \sum_{\tau\in T_M^{(c)}} \boldsymbol{\mu}(\tau, \boldsymbol{s}(\tau)), \tag{2.17}$$

$$\hat{\boldsymbol{\mu}}_{\text{av}}^{(b)}(M) \triangleq \frac{1}{M} \sum_{\tau\in T_M^{(b)}} \hat{\boldsymbol{\mu}}(\tau) = \frac{1}{M} \sum_{\tau\in T_M^{(b)}} \boldsymbol{\mu}(\tau) \otimes \mathbf{1}_{[\boldsymbol{\mu}(\tau)\leq \boldsymbol{s}(\tau)]}, \tag{2.18}$$

where vector $\mathbf{1}_{[\boldsymbol{\mu}(\tau)\leq \boldsymbol{s}(\tau)]}$ is defined before Section 2.1. Combining (2.13)(2.17)(2.18) yields

$$\hat{\boldsymbol{\mu}}_{\text{av}}(M) = \hat{\boldsymbol{\mu}}_{\text{av}}^{(c)}(M) + \hat{\boldsymbol{\mu}}_{\text{av}}^{(b)}(M). \tag{2.19}$$

Define

$$P_{\text{av}}^{(c)}(M) \triangleq \frac{1}{M} \sum_{\tau \in T_M^{(c)}} \sum_{n=1}^{N} P_n(\tau), \tag{2.20}$$

$$P_{\text{av}}^{(b)}(M) \triangleq \frac{1}{M} \sum_{\tau \in T_M^{(b)}} \sum_{n=1}^{N} P_n(\tau), \tag{2.21}$$

and $P_{\text{av}}(M)$ in (2.14) satisfies

$$P_{\text{av}}(M) = P_{\text{av}}^{(c)}(M) + P_{\text{av}}^{(b)}(M). \tag{2.22}$$

Consider the rate-power vector $(\hat{\boldsymbol{\mu}}_{\text{av}}^{(c)}(M); P_{\text{av}}^{(c)}(M))$ associated with channel-aware transmissions. From (2.17)(2.20), we have

$$\begin{aligned}
(\hat{\boldsymbol{\mu}}_{\text{av}}^{(c)}(M); P_{\text{av}}^{(c)}(M)) &= \frac{1}{M} \sum_{\tau \in T_M^{(c)}} \left(\boldsymbol{\mu}(\tau, \boldsymbol{s}(\tau)); \sum_{n=1}^{N} P_n(\tau) \right) \\
&\overset{(a)}{=} \frac{|T_M^{(c)}|}{M} \sum_{\boldsymbol{s} \in \mathcal{S}^N} \frac{|T_M^{(c)}(\boldsymbol{s})|}{|T_M^{(c)}|} \frac{1}{|T_M^{(c)}(\boldsymbol{s})|} \sum_{\tau \in T_M^{(c)}(\boldsymbol{s})} \left(\boldsymbol{\mu}(\tau, \boldsymbol{s}); \sum_{n=1}^{N} P_n(\tau) \right) \\
&= \gamma_M \sum_{\boldsymbol{s} \in \mathcal{S}^N} \sigma_M(\boldsymbol{s})\, \boldsymbol{x}_M(\boldsymbol{s}),
\end{aligned} \tag{2.23}$$

where (a) is from regrouping terms according to the observed channel states $\boldsymbol{s}$, $T_M^{(c)}(\boldsymbol{s}) \subset T_M^{(c)}$ is the subset of slots in which channel states $\boldsymbol{s}$ are probed, and

$$\gamma_M \triangleq \frac{|T_M^{(c)}|}{M}, \ \ \sigma_M(\boldsymbol{s}) \triangleq \frac{|T_M^{(c)}(\boldsymbol{s})|}{|T_M^{(c)}|}, \tag{2.24}$$

$$\boldsymbol{x}_M(\boldsymbol{s}) \triangleq \frac{1}{|T_M^{(c)}(\boldsymbol{s})|} \sum_{\tau \in T_M^{(c)}(\boldsymbol{s})} \left(\boldsymbol{\mu}(\tau, \boldsymbol{s}); \sum_{n=1}^{N} P_n(\tau) \right). \tag{2.25}$$

Observe that $\boldsymbol{x}_M(\boldsymbol{s})$ is a convex combination of vectors of the form $(\boldsymbol{\mu}(\tau,\boldsymbol{s});\sum_{n=1}^N P_n(\tau))$. By regrouping terms in (2.25) according to the value of $\boldsymbol{\mu}(\tau,\boldsymbol{s})$ and using (2.2), there exist real numbers $\{\alpha_M(\boldsymbol{\omega},\boldsymbol{s})\}_{\boldsymbol{\omega}\in\Omega(\boldsymbol{s})}$, where $\alpha_M(\boldsymbol{\omega},\boldsymbol{s}) \geq 0$ for all $\boldsymbol{\omega} \in \Omega(\boldsymbol{s})$ and $\sum_{\boldsymbol{\omega}\in\Omega(\boldsymbol{s})} \alpha_M(\boldsymbol{\omega},\boldsymbol{s}) = 1$, such that $\boldsymbol{x}_M(\boldsymbol{s})$ is rewritten as

$$\boldsymbol{x}_M(\boldsymbol{s}) = \sum_{\boldsymbol{\omega}\in\Omega(\boldsymbol{s})} \alpha_M(\boldsymbol{\omega},\boldsymbol{s}) \left(\boldsymbol{\omega}; P_{\text{meas}} + \sum_{n=1}^N 1_{[\omega_n>0]} P_{\text{tran}}\right). \tag{2.26}$$

We note that $\{\alpha_M(\boldsymbol{\omega},\boldsymbol{s})\}_{\boldsymbol{\omega}\in\Omega(\boldsymbol{s})}$ can be viewed as a probability distribution.

Likewise, from (2.18)(2.21), the rate-power vector $(\hat{\boldsymbol{\mu}}_{\text{av}}^{(b)}(M); P_{\text{av}}^{(b)}(M))$ associated with channel-blind transmissions satisfies

$$\begin{aligned}
(\hat{\boldsymbol{\mu}}_{\text{av}}^{(b)}(M); P_{\text{av}}^{(b)}(M)) &= \frac{1}{M} \sum_{\tau\in T_M^{(b)}} \left(\boldsymbol{\mu}(\tau) \otimes \mathbf{1}_{[\boldsymbol{\mu}(\tau)\leq \boldsymbol{s}(\tau)]}; \sum_{n=1}^N P_n(\tau)\right) \\
&\overset{(a)}{=} \frac{1}{M} \sum_{\boldsymbol{\omega}\in\Omega} \sum_{\tau\in T_M^{(b)}(\boldsymbol{\omega})} \left(\boldsymbol{\omega} \otimes \mathbf{1}_{[\boldsymbol{\omega}\leq \boldsymbol{s}(\tau)]}; \sum_{n=1}^N P_n(\tau)\right) \\
&\overset{(b)}{=} (1-\gamma_M) \sum_{\boldsymbol{\omega}\in\Omega} \beta_M(\boldsymbol{\omega})\ \boldsymbol{y}_M(\boldsymbol{\omega}),
\end{aligned} \tag{2.27}$$

where (a) is from regrouping terms according to the value of $\boldsymbol{\mu}(\tau)$, $T_M^{(b)}(\boldsymbol{\omega}) \subset T_M^{(b)}$ denotes the subset of slots in which $\boldsymbol{\mu}(\tau) = \boldsymbol{\omega}$ is allocated, and (b) follows

$$\begin{gathered}
1-\gamma_M = \frac{|T_M^{(b)}|}{M}, \quad \beta_M(\boldsymbol{\omega}) \triangleq \frac{|T_M^{(b)}(\boldsymbol{\omega})|}{|T_M^{(b)}|}, \\
\boldsymbol{y}_M(\boldsymbol{\omega}) \triangleq \frac{1}{|T_M^{(b)}(\boldsymbol{\omega})|} \sum_{\tau\in T_M^{(b)}(\boldsymbol{\omega})} \left(\boldsymbol{\omega} \otimes \mathbf{1}_{[\boldsymbol{\omega}\leq \boldsymbol{s}(\tau)]}; \sum_{n=1}^N P_n(\tau)\right).
\end{gathered} \tag{2.28}$$

Note that $\{\beta_M(\boldsymbol{\omega})\}_{\boldsymbol{\omega}\in\Omega}$ can also be viewed as a probability distribution.

Combining (2.19) (2.22) (2.23) (2.27) yields

$$(\hat{\boldsymbol{\mu}}_{\text{av}}(M); P_{\text{av}}(M)) = \gamma_M \sum_{\boldsymbol{s}\in\mathcal{S}^N} \sigma_M(\boldsymbol{s})\ \boldsymbol{x}_M(\boldsymbol{s}) + (1-\gamma_M) \sum_{\boldsymbol{\omega}\in\Omega} \beta_M(\boldsymbol{\omega})\ \boldsymbol{y}_M(\boldsymbol{\omega}), \tag{2.29}$$

where $\boldsymbol{x}_M(\boldsymbol{s})$ and $\boldsymbol{y}_M(\boldsymbol{\omega})$ are shown in (2.26) and (2.28), respectively. In (2.29), all sequences $\{\gamma_M\}_M$, $\{\sigma_M(\boldsymbol{s})\}_M$, $\{\alpha_M(\boldsymbol{\omega}, \boldsymbol{s})\}_M$, $\{\beta_M(\boldsymbol{\omega})\}_M$, $\{\boldsymbol{y}_M(\boldsymbol{\omega})\}_M$ are bounded. By iteratively applying Weierstrass's Theorem, there exists a subsequence $\{M_k\}_k \subset \{M_n\}_n$, where $\{M_n\}_n$ is given in (2.16), such that all the above sequences have convergent subsequences indexed by $\{M_k\}_k$. It follows that there exists a scalar $\gamma \in [0, 1]$, a probability distribution $\{\alpha(\boldsymbol{\omega}, \boldsymbol{s})\}_{\boldsymbol{\omega}\in\Omega(\boldsymbol{s})}$ for each $\boldsymbol{s} \in \mathcal{S}^N$, and a probability distribution $\{\beta(\boldsymbol{\omega})\}_{\boldsymbol{\omega}\in\Omega}$, such that, as $k \to \infty$,

$$
\begin{aligned}
\gamma_{M_k} &\to \gamma, \\
\alpha_{M_k}(\boldsymbol{\omega}, \boldsymbol{s}) &\to \alpha(\boldsymbol{\omega}, \boldsymbol{s}), \quad \text{for all } \boldsymbol{s} \in \mathcal{S}^N \text{ and } \boldsymbol{\omega} \in \Omega(\boldsymbol{s}), \\
\beta_{M_k}(\boldsymbol{\omega}) &\to \beta(\boldsymbol{\omega}), \quad \text{for all } \boldsymbol{\omega} \in \Omega.
\end{aligned}
$$

Let $\pi_{\boldsymbol{s}}$ be the steady state probability of channel state vector $\boldsymbol{s}$. Because channel probing in a slot is independent of channel states in that slot, we have $\sigma_{M_k}(\boldsymbol{s}) \to \pi_{\boldsymbol{s}}$ as $k \to \infty$ by the Law of Large Numbers (LLN). For each $\boldsymbol{\omega} \in \Omega$, the vectors $\boldsymbol{\omega} \otimes \mathbf{1}_{[\boldsymbol{\omega} \le \boldsymbol{s}(\tau)]}$ are i.i.d. over slots $\tau \in T^{(b)}_{M_k}(\boldsymbol{\omega})$. Thus, by LLN,

$$
\boldsymbol{y}_{M_k}(\boldsymbol{\omega}) \to \left(\boldsymbol{\omega} \otimes \boldsymbol{Pr}(\boldsymbol{S} \ge \boldsymbol{\omega}); \sum_{n=1}^{N} 1_{[\omega_n > 0]} P_{\text{tran}}\right), \quad \text{as } k \to \infty.
$$

From the above analysis, as $k \to \infty$, the subsequence $\{(\hat{\boldsymbol{\mu}}_{\text{av}}(M_k); P_{\text{av}}(M_k))\}$ converges to

$$
\begin{aligned}
&\gamma \sum_{\boldsymbol{s}\in\mathcal{S}^N} \pi_{\boldsymbol{s}} \left[\sum_{\boldsymbol{\omega}\in\Omega(\boldsymbol{s})} \alpha(\boldsymbol{\omega}, \boldsymbol{s}) \left(\boldsymbol{\omega}; P_{\text{meas}} + \sum_{n=1}^{N} 1_{[\omega_n > 0]} P_{\text{tran}}\right)\right] \\
&+ (1-\gamma) \sum_{\boldsymbol{\omega}\in\Omega} \beta(\boldsymbol{\omega}) \left(\boldsymbol{\omega} \otimes \boldsymbol{Pr}(\boldsymbol{S} \ge \boldsymbol{\omega}); \sum_{n=1}^{N} 1_{[\omega_n > 0]} P_{\text{tran}}\right).
\end{aligned} \tag{2.30}
$$

Using the stability definition and Lemma 1 in Neely et al. [2003], necessarily we have $\boldsymbol{\lambda} \leq \liminf_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \hat{\boldsymbol{\mu}}(\tau)$ with probability 1. It yields that

$$\begin{aligned}
\boldsymbol{\lambda} &\overset{(a)}{\leq} \lim_{k\to\infty} \frac{1}{M_k} \sum_{\tau=0}^{M_k-1} \hat{\boldsymbol{\mu}}(\tau) \\
&= \lim_{k\to\infty} \hat{\boldsymbol{\mu}}_{\text{av}}(M_k) \\
&= \gamma \sum_{\boldsymbol{s}\in\mathcal{S}^N} \pi_{\boldsymbol{s}} \left(\sum_{\boldsymbol{\omega}\in\Omega(\boldsymbol{s})} \alpha(\boldsymbol{\omega},\boldsymbol{s})\,\boldsymbol{\omega} \right) + (1-\gamma) \sum_{\boldsymbol{\omega}\in\Omega} \beta(\boldsymbol{\omega})\left(\boldsymbol{\omega} \otimes \boldsymbol{Pr}(\boldsymbol{S} \geq \boldsymbol{\omega})\right),
\end{aligned} \tag{2.31}$$

where (a) follows that lim inf of a sequence is the infimum of all limit points in that sequence. Combining (2.16) and (2.30) yields

$$\begin{aligned}
\overline{P}_{\text{av}} = \gamma \sum_{\boldsymbol{s}\in\mathcal{S}^N} \pi_{\boldsymbol{s}} &\left[\sum_{\boldsymbol{\omega}\in\Omega(\boldsymbol{s})} \alpha(\boldsymbol{\omega},\boldsymbol{s}) \left(P_{\text{meas}} + \sum_{n=1}^{N} 1_{[\omega_n>0]} P_{\text{tran}} \right) \right] \\
&+ (1-\gamma) \sum_{\boldsymbol{\omega}\in\Omega} \beta(\boldsymbol{\omega}) \left(\sum_{n=1}^{N} 1_{[\omega_n>0]} P_{\text{tran}} \right).
\end{aligned} \tag{2.32}$$

From (2.31) (2.32), if there is a $\boldsymbol{\lambda}$-stabilizing policy Φ, there is a stationary randomized policy $\hat{\Phi}$ operating as follows: In every slot, $\hat{\Phi}$ probes all channels with probability γ. If the channel state vector $\boldsymbol{s}$ is acquired, $\hat{\Phi}$ allocates channel-aware rate vector $\boldsymbol{\omega} \in \Omega(\boldsymbol{s})$ with probability $\alpha(\boldsymbol{\omega},\boldsymbol{s})$. Otherwise, $\hat{\Phi}$ blindly allocates rate vector $\boldsymbol{\omega} \in \Omega$ with probability $\beta(\boldsymbol{\omega})$. Policy $\hat{\Phi}$ yields a service rate vector greater than or equal to $\boldsymbol{\lambda}$ entrywise, and consumes average power $\overline{P}_{\text{av}}$. This finishes the necessity part of the proof.

(Sufficiency) Conversely, for each rate vector $\boldsymbol{\lambda}$ interior to Λ, there exists a scalar $\epsilon > 0$ such that $\boldsymbol{\lambda} + \boldsymbol{\epsilon}$ is interior to Λ, where $\boldsymbol{\epsilon}$ is an all-ϵ vector. The optimal solution to $\mathfrak{P}(\boldsymbol{\lambda}+\boldsymbol{\epsilon})$ yields a stationary randomized policy $\hat{\Phi}$ whose average service rate vector is greater than or equal to $\boldsymbol{\lambda}+\boldsymbol{\epsilon}$ entrywise. By [Georgiadis et al. 2006, Lemma 3.6], policy $\hat{\Phi}$ stabilizes $\boldsymbol{\lambda}$ with average power $P_{\text{opt}}(\boldsymbol{\lambda}+\boldsymbol{\epsilon})$. As $\epsilon \to 0$, there exists a stationary randomized policy supporting $\boldsymbol{\lambda}$ with average power arbitrarily close to $P_{\text{opt}}(\boldsymbol{\lambda})$. □

2.9.4 Proof of Theorem 2.11

Taking the square of (2.3) for each n and using the facts that

$$(\max(Q_n(t) - \hat{\mu}_n(t), 0))^2 \leq (Q_n(t) - \hat{\mu}_n(t))^2,$$
$$Q_n(t) - \hat{\mu}_n(t) \leq Q_n(t),\ \hat{\mu}_n(t) \leq \mu_{\max},\ a_n(t) \leq A_{\max},$$

we have

$$\sum_{n=1}^{N} \left(Q_n^2(t+1) - Q_n^2(t)\right) \leq B - 2\sum_{n=1}^{N} Q_n(t)(\hat{\mu}_n(t) - a_n(t)), \tag{2.33}$$

where $B \triangleq (\mu_{max}^2 + A_{max}^2)N$. We define the Lyapunov function $L(t) \triangleq \sum_{n=1}^{N} Q_n^2(t)$ and the one-step Lyapunov drift $\Delta(\boldsymbol{Q}(t)) \triangleq \mathbb{E}\left[L(t+1) - L(t) | \boldsymbol{Q}(t)\right]$. By taking expectation of (2.33) conditioning on current backlog $\boldsymbol{Q}(t)$ and noting that arrival processes are i.i.d. over slots, we have

$$\Delta(\boldsymbol{Q}(t)) \leq B + 2\sum_{n=1}^{N} Q_n(t)\,\lambda_n - \sum_{n=1}^{N} 2\,Q_n(t)\,\mathbb{E}\left[\hat{\mu}_n(t) \mid \boldsymbol{Q}(t)\right]. \tag{2.34}$$

Adding the weighted power cost $V\mathbb{E}\left[\sum_{n=1}^{N} P_n(t) \mid \boldsymbol{Q}(t)\right]$ to both sides of (2.34) yields

$$\begin{aligned}\Delta(\boldsymbol{Q}(t)) + V\mathbb{E}\left[\sum_{n=1}^{N} P_n(t) \mid \boldsymbol{Q}(t)\right] &\leq B + 2\sum_{n=1}^{N} Q_n(t)\,\lambda_n \\ -\left(\sum_{n=1}^{N} 2\,Q_n(t)\,\mathbb{E}\left[\hat{\mu}_n(t) \mid \boldsymbol{Q}(t)\right] - V\mathbb{E}\left[\sum_{n=1}^{N} P_n(t) \mid \boldsymbol{Q}(t)\right]\right)&. \end{aligned} \tag{2.35}$$

The DCA algorithm is designed to minimize the right side of (2.35), i.e., to maximize $f(\boldsymbol{Q}(t), \boldsymbol{\chi}(t))$ in (2.4), over all feasible controls $\boldsymbol{\chi}(t)$ in every slot. The intuition is as follows. Minimizing $\Delta(\boldsymbol{Q}(t))$, the expected growth of queue sizes over a slot, stabilizes the network. At the same time, we want to decrease the power consumption. These two goals create a tradeoff because queue sizes can be decreased by consuming more power, such as probing channels and allocating channel-aware rates in every slot. Therefore, it is natural to minimize a weighted sum of them, which is the left side of (2.35). The V parameter in the weighted sum controls the tradeoff between queue stability and power. The DCA algorithm seeks to minimize the weighted sum by minimizing its upper bound.

Next, we analyze the performance of DCA. By definition of DCA, the right side of (2.35) under DCA is less than or equal to that under the optimal stationary randomized policy, denoted by Φ^r, associated with the optimal solution to the problem $\mathfrak{P}(\boldsymbol{\lambda}+\boldsymbol{\epsilon})$ in Theorem 2.7, where the vector $\boldsymbol{\epsilon} > \mathbf{0}$ is an all-ϵ vector such that $\boldsymbol{\lambda}+\boldsymbol{\epsilon}$ is interior to Λ. Let $\hat{\boldsymbol{\mu}}^r(t) = (\hat{\mu}_n^r(t))_{n=1}^N$ and $(P_1^r(t), \ldots, P_N^r(t))$ be the effective service rate vector and power consumption of policy Φ^r in slot t. Consequently, (2.35) under DCA is further upper bounded as

$$\Delta(\boldsymbol{Q}(t)) + V\mathbb{E}\left[\sum_{n=1}^N P_n(t) \mid \boldsymbol{Q}(t)\right] \leq B + 2\sum_{n=1}^N Q_n(t)\,\lambda_n \\ - \left(\sum_{n=1}^N 2\,Q_n(t)\,\mathbb{E}\left[\hat{\mu}_n^r(t) \mid \boldsymbol{Q}(t)\right] - V\mathbb{E}\left[\sum_{n=1}^N P_n^r(t) \mid \boldsymbol{Q}(t)\right]\right). \tag{2.36}$$

Since policy Φ^r does not use queue backlog information $\boldsymbol{Q}(t)$, Corollary 2.8 shows that

$$\mathbb{E}\left[\hat{\boldsymbol{\mu}}^r(t) \mid \boldsymbol{Q}(t)\right] \geq \boldsymbol{\lambda}+\boldsymbol{\epsilon}, \tag{2.37}$$

$$\mathbb{E}\left[\sum_{n=1}^N P_n^r(t) \mid \boldsymbol{Q}(t)\right] = P_{\text{opt}}(\boldsymbol{\lambda}+\boldsymbol{\epsilon}) \tag{2.38}$$

in every slot t. Plugging (2.37) and (2.38) into (2.36) yields

$$\Delta(\boldsymbol{Q}(t)) + V\mathbb{E}\left[\sum_{n=1}^N P_n(t) \mid \boldsymbol{Q}(t)\right] \leq B - 2\epsilon\sum_{n=1}^N Q_n(t) + V P_{\text{opt}}(\boldsymbol{\lambda}+\boldsymbol{\epsilon}). \tag{2.39}$$

Taking expectation of (2.39) over $\boldsymbol{Q}(t)$, summing it from $t=0$ to $\tau-1$, dividing the sum by τ, and noting $\boldsymbol{Q}(0)=\mathbf{0}$ and $\mathbb{E}\left[L(\boldsymbol{Q}(\tau))\right] \geq 0$, we have

$$\frac{2\epsilon}{\tau}\sum_{t=0}^{\tau-1}\sum_{n=1}^N \mathbb{E}\left[Q_n(t)\right] \leq B + V P_{\text{opt}}(\boldsymbol{\lambda}+\boldsymbol{\epsilon}) - \frac{V}{\tau}\mathbb{E}\left[\sum_{t=0}^{\tau-1}\sum_{n=1}^N P_n(t)\right]. \tag{2.40}$$

Ignoring the last term of (2.40), dividing by 2ϵ, using $P_{\text{opt}}(\boldsymbol{\lambda}+\boldsymbol{\epsilon}) \leq P_{\text{meas}} + NP_{\text{tran}}$, and taking a lim sup as $\tau \to \infty$, we get

$$\limsup_{\tau\to\infty} \frac{1}{\tau} \sum_{t=0}^{\tau-1} \sum_{n=1}^{N} \mathbb{E}\left[Q_n(t)\right] \leq \frac{B + V(P_{\text{meas}} + NP_{\text{tran}})}{2\epsilon}. \tag{2.41}$$

Ignoring the first term of (2.40), dividing the result by V, and taking a lim sup as $\tau \to \infty$, we get

$$\limsup_{\tau\to\infty} \frac{1}{\tau} \sum_{t=0}^{\tau-1} \sum_{n=1}^{N} \mathbb{E}\left[P_n(t)\right] \leq \frac{B}{V} + P_{\text{opt}}(\boldsymbol{\lambda}+\boldsymbol{\epsilon}). \tag{2.42}$$

(2.41) and (2.42) hold for any $\epsilon > 0$ such that $\boldsymbol{\lambda}+\boldsymbol{\epsilon}$ is interior to the network capacity region Λ. We can tighten the bounds by: (1) setting $\epsilon = \epsilon_{\max}$ in (2.41), where $\epsilon_{\max} > 0$ is the largest real number satisfying $\boldsymbol{\lambda} + \boldsymbol{\epsilon}_{\max} \in \Lambda$; (2) setting $\epsilon = 0$ in (2.42). It follows that

$$\limsup_{\tau\to\infty} \frac{1}{\tau} \sum_{t=0}^{\tau-1} \sum_{n=1}^{N} \mathbb{E}\left[Q_n(t)\right] \leq \frac{B + V(P_{\text{meas}} + NP_{\text{tran}})}{2\epsilon_{\max}},$$
$$\limsup_{\tau\to\infty} \frac{1}{\tau} \sum_{t=0}^{\tau-1} \sum_{n=1}^{N} \mathbb{E}\left[P_n(t)\right] \leq \frac{B}{V} + P_{\text{opt}}(\boldsymbol{\lambda}).$$

□

Chapter 3

Exploiting Wireless Channel Memory

Much previous work on wireless opportunistic scheduling, including our work on dynamic channel probing in Chapter 2, assumes channels are i.i.d. over slots. Thus, the channel memory associated with ergodic (non-i.i.d.) systems is often overlooked. The main reasons are: (1) the i.i.d. channel assumption is easier to deal with analytically; (2) under the assumption that channel states are instantaneously known, throughput-optimal policies developed under the i.i.d. assumption can as well be shown optimal for general ergodic channels [Neely et al. 2005, Tassiulas 1997]. This chapter explores how to exploit channel memory in cases when channel states are not known in advance.

Specifically, we consider a time-slotted wireless base station that serves multiple users over independent Markov ON/OFF channels.[1] Every channel is assumed positively correlated, so that an ON state is more likely followed by another ON state in the next slot. We consider the scenario that channels are never probed, so that their instantaneous states are never known. On each slot, the base station transmits a packet to a chosen user, and receives an ACK/NACK feedback. The transmission succeeds if the channel is ON, and fails otherwise. Since channels are ON/OFF, the ACK/NACK feedback uncovers the state of the used channel in the previous slot, and provides partial information of future states on that channel. The user selection decision on each slot may take advantage of

[1] It has been shown that wireless channels can be adequately modeled as Markov chains [Wang and Chang 1996, Zorzi et al. 1996], especially in high-speed transmission regimes. Since each time slot comprises a short period of time, channel states are likely correlated across slots.

this partial channel information. We study: (1) What is the network capacity region? (2) What are the throughput-optimal policies?

As discussed in Section 1.3.1, the network capacity region in this problem is difficult to compute because it is the same as characterizing the set of all achievable time average reward vectors in a restless multi-armed bandit system. Our alternative approach in this chapter is to explore the structure of the Markovian channels to construct a good inner bound on the network capacity region by randomizing well-designed round robin policies. We will construct an outer capacity bound to study the tightness of the inner capacity region. Specifically, in the case of symmetric channels and a large number of users, we show that, as data rates are more *balanced*, or in a geometric sense as the direction of the data rate vector in the Euclidean space is closer to the 45-degree angle, the inner bound converges geometrically fast to the outer bound as the number of channels increases, and both bounds are tight. The outer bound comes from analyzing a fictitious channel model in which every scheduling policy yields higher throughput than it does in the real network.

This chapter is organized as follows. The network model is given in Section 3.1. Section 3.2 presents a carefully designed round robin policy that is the fundamental building block of our inner capacity bound. Our inner and outer capacity bounds are constructed in Sections 3.3, and compared in Section 3.4 in the case of symmetric channels. Section 3.5 constructs a simple queue-dependent round robin policy that is throughput-optimal over the inner network capacity region. A novel frame-based Lyapunov drift analysis is used to design the policy and analyze its performance.

3.1 Network Model

We consider a base station transmitting data to N users through N Markov ON/OFF channels. Suppose time is slotted with normalized slots $t \in \mathbb{Z}^+$. Each channel $n \in \{1, \dots, N\}$ is modeled as a two-state ON/OFF Markov chain (see Fig. 3.1). Let $s_n(t) \in$

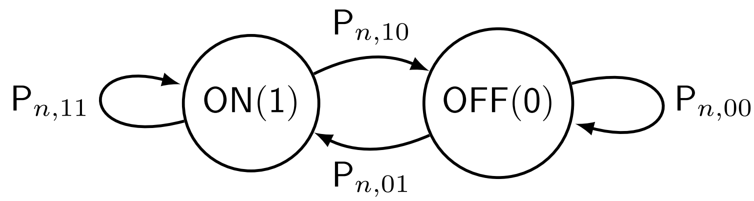

Figure 3.1: A two-state Markov ON/OFF chain for channel $n \in \{1, 2, \dots, N\}$.

$\{\mathsf{OFF}, \mathsf{ON}\}$ denote the state of channel n in slot t. The state $s_n(t)$ of channel n evolves according to the transition probability matrix

$$\mathbf{P}_n = \begin{bmatrix} \mathsf{P}_{n,00} & \mathsf{P}_{n,01} \\ \mathsf{P}_{n,10} & \mathsf{P}_{n,11} \end{bmatrix},$$

where state ON is represented by 1 and OFF by 0, and $\mathsf{P}_{n,ij}$ denotes the transition probability from state i to j. We assume $\mathsf{P}_{n,11} < 1$ for all channels n so that no channel is constantly ON (incorporating constantly ON channels like wired links is easy and omitted). We suppose channel states are fixed in every slot and may only change at slot boundaries. We assume all channels are positively correlated, which, in terms of transition probabilities, is equivalent to assuming $\mathsf{P}_{n,11} > \mathsf{P}_{n,01}$ or $\mathsf{P}_{n,01} + \mathsf{P}_{n,10} < 1$ for all channels n.[2] We suppose the base station keeps N queues of infinite capacity to store exogenous packet arrivals destined for the N users. At the beginning of every slot, the base station attempts to transmit a packet (if there is any) to a selected user. We suppose the base station has no channel probing capability and must select users oblivious of the current channel states. If a user is selected and its current channel state is ON, one packet is successfully delivered to that user. Otherwise, the transmission fails and zero packets are served. At the end of a slot in which the base station serves a user,

[2] The assumption $\mathsf{P}_{n,11} > \mathsf{P}_{n,01}$ yields that the state $s_n(t)$ of channel n has auto-covariance $\mathbb{E}\left[(s_n(t) - \mathbb{E}s_n(t))(s_n(t+1) - \mathbb{E}s_n(t+1))\right] > 0$. The case $\mathsf{P}_{n,11} = \mathsf{P}_{n,01}$ corresponds to a channel having i.i.d. states over slots. Although we can naturally incorporate i.i.d. channels into our model and all our results still hold, we exclude that case because we shall show how throughput can be improved by channel memory, which i.i.d. channels do not have. The degenerate case where all channels are i.i.d. over slots is fully solved in Li and Neely [2010a] or in Chapter 2.

an ACK/NACK message is fed back from the selected user to the base station through an independent error-free control channel, according to whether the transmission succeeds. Failing to receive an ACK is regarded as a NACK. Since channel states are either ON or OFF, such feedback reveals the channel state of the selected user in the last slot.

We define the N-dimensional *information state vector* $\boldsymbol{\omega}(t) = (\omega_n(t))_{n=1}^{N}$ where $\omega_n(t)$ is the probability that channel n is ON in slot t conditioning on channel observation history. In other words,

$$\omega_n(t) \triangleq \Pr\left[s_n(t) = \mathsf{ON} \mid \text{all past observations of channel } n\right].$$

We assume initially $\omega_n(0) = \pi_{n,\mathsf{ON}}$ for all n, where $\pi_{n,\mathsf{ON}}$ denotes the stationary probability that channel n is ON. The vector $\boldsymbol{\omega}(t)$ is known to be a *sufficient statistic* [Bertsekas 2005, Chapter 5.4]. That is, instead of tracking the whole system history, the base station can act optimally based only on $\boldsymbol{\omega}(t)$. The base station shall keep track of the information state process $\{\boldsymbol{\omega}(t)\}_{t\in\mathbb{Z}^+}$.

Throughout the chapter, we assume that the transition probability matrices $\mathbf{P}_n$ of all channels are known to the base station. In practice, the matrix $\mathbf{P}_n$ for channel n may be learned in an initial training period, in which the base station continuously transmits packets over channel n in every slot. In this period we compute a sample average $\overline{Y_n}$ of the durations $(Y_{n,1}, Y_{n,2}, Y_{n,3}, \ldots)$ that channel n is continuously ON. It is easy to see that $Y_{n,k}$ are i.i.d. over k with $\mathbb{E}\left[Y_{n.k}\right] = 1/\mathsf{P}_{n,10}$. As a result, we may use $1/\overline{Y_n}$ as an estimate of $\mathsf{P}_{n,10}$. The transition probability $\mathsf{P}_{n,01}$ can be estimated similarly. This estimation method works when channels are stationary.

Next, let $n(t) \in \{1, 2, \ldots, N\}$ be the user served in slot t. Based on the ACK/NACK feedback, the information state vector $\boldsymbol{\omega}(t)$ is updated in every slot according to:

$$\omega_n(t+1) = \begin{cases} \mathsf{P}_{n,01}, & \text{if } n = n(t),\ s_n(t) = \mathsf{OFF} \\ \mathsf{P}_{n,11}, & \text{if } n = n(t),\ s_n(t) = \mathsf{ON} \\ \omega_n(t)\mathsf{P}_{n,11} + (1-\omega_n(t))\mathsf{P}_{n,01}, & \text{if } n \neq n(t) \end{cases}, \ \forall\, n \in \{1, 2, \ldots, N\}. \quad (3.1)$$

If in the most recent use of channel n, we observed (through feedback) that its state was $i \in \{0, 1\}$ in slot $(t-k)$ for some $k \leq t$, then $\omega_n(t)$ is equal to the k-step transition probability $\mathsf{P}^{(k)}_{n,i1}$. In general, for any fixed n, the information state $\omega_n(t)$ takes values in the countably infinite set $\mathcal{W}_n = \{\mathsf{P}^{(k)}_{n,01}, \mathsf{P}^{(k)}_{n,11} : k \in \mathbb{N}\} \cup \{\pi_{n,\mathsf{ON}}\}$. By eigenvalue decomposition on $\mathbf{P}_n$ [Gallager 1996, Chapter 4], the k-step transition probability matrix $\mathbf{P}^{(k)}_n$ is

$$\begin{aligned} \mathbf{P}^{(k)}_n &\triangleq \begin{bmatrix} \mathsf{P}^{(k)}_{n,00} & \mathsf{P}^{(k)}_{n,01} \\ \mathsf{P}^{(k)}_{n,10} & \mathsf{P}^{(k)}_{n,11} \end{bmatrix} \\ &= (\mathbf{P}_n)^k \\ &= \frac{1}{x_n} \begin{bmatrix} \mathsf{P}_{n,10} + \mathsf{P}_{n,01}(1-x_n)^k & \mathsf{P}_{n,01}\left(1-(1-x_n)^k\right) \\ \mathsf{P}_{n,10}(1-(1-x_n)^k) & \mathsf{P}_{n,01} + \mathsf{P}_{n,10}(1-x_n)^k \end{bmatrix}, \end{aligned} \quad (3.2)$$

where we have defined $x_n \triangleq \mathsf{P}_{n,01} + \mathsf{P}_{n,10}$. Assuming that channels are positively correlated, i.e., $x_n < 1$, we have the following lemma as an immediate result from (3.2).

Lemma 3.1. For a positively correlated ($\mathsf{P}_{n,11} > \mathsf{P}_{n,01}$) Markov ON/OFF channel with transition probability matrix $\mathbf{P}_n$, we have:

1. The stationary probability $\pi_{n,\mathsf{ON}} = \mathsf{P}_{n,01}/x_n$.
2. The k-step transition probability $\mathsf{P}^{(k)}_{n,01}$ is nondecreasing in k and $\mathsf{P}^{(k)}_{n,11}$ nonincreasing in k. Both $\mathsf{P}^{(k)}_{n,01}$ and $\mathsf{P}^{(k)}_{n,11}$ converge to $\pi_{n,\mathsf{ON}}$ as $k \to \infty$.

As a corollary of Lemma 3.1, we have

$$\mathsf{P}_{n,11} \geq \mathsf{P}_{n,11}^{(k_1)} \geq \mathsf{P}_{n,11}^{(k_2)} \geq \pi_{n,\mathsf{ON}} \geq \mathsf{P}_{n,01}^{(k_3)} \geq \mathsf{P}_{n,01}^{(k_4)} \geq \mathsf{P}_{n,01} \tag{3.3}$$

for any integers $k_1 \leq k_2$ and $k_3 \geq k_4$ (see Fig. 3.2). To maximize network throughput, (3.3) has some fundamental implications. First, noting that $\omega_n(t)$ represents the transmission success probability over channel n in slot t, we shall keep serving a channel whenever its information state is $\mathsf{P}_{n,11}$, because $\mathsf{P}_{n,11}$ is the best state possible. Second, given that a channel was OFF in its last use, its information state improves as long as the channel remains idle. This shows that we shall wait as long as possible before reusing such a channel. Indeed, when channels are symmetric (i.e., all channels are independent and have the same statistics $\mathbf{P}_n = \mathbf{P}$ for all n), Ahmad et al. [2009] shows that a myopic round robin policy with these properties maximizes the sum throughput of the network.

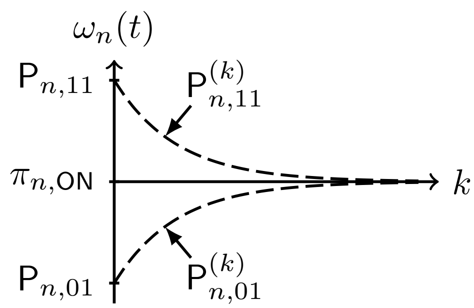

Figure 3.2: The k-step transition probabilities $\mathsf{P}_{n,01}^{(k)}$ and $\mathsf{P}_{n,11}^{(k)}$ of a positively correlated Markov ON/OFF channel.

3.2 Round Robin Policy $\mathsf{RR(M)}$

For any integer $M \in \{1, 2, \ldots, N\}$, we present a special round robin policy $\mathsf{RR(M)}$ that serves a subset of M users. We label these users by $\{1, \ldots, M\}$, and serve them in the circular order $1 \to 2 \to \cdots \to M \to 1 \to \cdots$. In general, the $\mathsf{RR(M)}$ policy can work on any subset of M users with any pre-determined ordering within the subset. This $\mathsf{RR(M)}$ policy is in the same spirit as the myopic round robin policy in Ahmad et al. [2009] that maximizes sum throughput over symmetric channels, but is carefully designed to have closed-form performance that is easy to compute.[3] The $\mathsf{RR(M)}$ policy preserves the same asymptotical throughput optimality as the optimal myopic policy in Ahmad et al. [2009], and is a fundamental building block in later analysis.

Round Robin Policy $\mathsf{RR(M)}$:

1. At time 0, the base station starts with channel 1. Suppose initially $\omega_n(0) = \pi_{n,\mathsf{ON}}$ for all n.

2. Suppose at time t, the base station switches to channel n. Then, it transmits a *data* packet to user n with probability $\mathsf{P}_{n,01}^{(M)}/\omega_n(t)$ and a *dummy* packet otherwise. In both cases, we receive ACK/NACK information at the end of the slot.

3. At time $(t+1)$, if a dummy packet is sent at time t, the base station switches to channel $(n \mod M)+1$ and go to Step 2. Otherwise, it keeps transmitting data packets over channel n until a NACK is received; then, it switches to channel $(n \mod M)+1$ and goes to Step 2.

4. Update $\boldsymbol{\omega}(t)$ according to (3.1) in every slot.

We note that Step 2 of $\mathsf{RR(M)}$ only makes sense if $\omega_n(t) \geq \mathsf{P}_{n,01}^{(M)}$ for all slots t that are the first slots after switching to a new channel. We prove that this inequality is always true in the next lemma.

Lemma 3.2 (Proof in Section 3.8.1)**.** Under $\mathsf{RR(M)}$, whenever the base station switches to channel $n \in \{1, 2, \ldots, M\}$ for another round of transmission, its current information state satisfies $\omega_n(t) \geq \mathsf{P}_{n,01}^{(M)}$.

[3] Over symmetric Markov $\mathsf{ON/OFF}$ channels, the sum throughput of the optimal myopic policy in Ahmad et al. [2009] is difficult to compute when N is large because it involves a high-order Markov chain.

We note that policy $\mathsf{RR(M)}$ is conservative and not throughput-optimal. For example, we can improve throughput by always sending data packets but no dummy ones. Also, it does not follow the guidelines we provide at the end of Section 3.1 for maximum throughput. Nonetheless, we will see later that, in the case of symmetric channels, the sum throughput under $\mathsf{RR(M)}$ is close to optimal when M is large. More importantly, the underlying analysis of $\mathsf{RR(M)}$ is tractable so that we can mix such round robin policies over different subsets of users to form a non-trivial inner bound on the network capacity region. The tractability of $\mathsf{RR(M)}$ is because it is equivalent to the following *fictitious* round robin policy (which can be proved as a corollary of Lemma 3.3 provided later).

Equivalent Fictitious Round Robin:

1. At time 0, start with channel 1.

2. When the base station switches to channel n, *set its current information state to* $\mathsf{P}_{n,01}^{(M)}$.[4] Keep transmitting data packets over channel n until a NACK is received. Then switch to channel $(n \mod M) + 1$ and repeat Step 2.

For any round robin policy that serves channels in the circular order $1 \to 2 \to \cdots \to M \to 1 \to \cdots$, the technique of resetting the information state to $\mathsf{P}_{n,01}^{(M)}$ creates a system with an information state that is *worse* than the information state under the actual system. To see this, since in the actual system channels are served in the circular order, after we switch away from serving a particular channel n, we serve the other $(M-1)$ channels for at least one slot each, and so we return to channel n after at least M slots. Thus, its starting information state is always at least $\mathsf{P}_{n,01}^{(M)}$ (the proof is similar to that of Lemma 3.2). Intuitively, since information states represent the success probabilities of packet transmissions, resetting them to lower values degrades throughput. This is the reason why our inner capacity bound constructed later using $\mathsf{RR(M)}$ provides a throughput lower bound for a large class of policies.

[4]In reality, we cannot *set* the information state of a channel, and therefore the policy is fictitious.

3.2.1 Throughput Analysis

We analyze the throughput vector achieved by $\mathsf{RR(M)}$. Under $\mathsf{RR(M)}$, let L_{kn} denote the duration of the kth time the base station stays with channel n. A sample path of the $\{L_{kn}\}$ process is

$$(\underbrace{L_{11}, L_{12}, \dots, L_{1M}}_{\text{round } k=1}, \underbrace{L_{21}, L_{22}, \dots, L_{2M}}_{\text{round } k=2}, L_{31}, \dots). \tag{3.4}$$

The next lemma presents useful properties of L_{kn}, which serve as the foundation of the throughput analysis in the rest of the chapter.

Lemma 3.3. For any integer k and $n \in \{1, 2, \dots, M\}$,

1. The probability mass function of L_{kn} is independent of k, and is

$$L_{kn} = \begin{cases} 1, & \text{with prob. } 1 - \mathsf{P}_{n,01}^{(M)} \\ j \geq 2, & \text{with prob. } \mathsf{P}_{n,01}^{(M)} (\mathsf{P}_{n,11})^{(j-2)} \, \mathsf{P}_{n,10}. \end{cases}$$

   As a result, for all $k \in \mathbb{N}$ we have

$$\mathbb{E}[L_{kn}] = 1 + \frac{\mathsf{P}_{n,01}^{(M)}}{\mathsf{P}_{n,10}} = 1 + \frac{\mathsf{P}_{n,01}(1-(1-x_n)^M)}{x_n \mathsf{P}_{n,10}}, \quad x_n = \mathsf{P}_{n,01} + \mathsf{P}_{n,10}. \tag{3.5}$$

2. The number of data packets served in L_{kn} is $(L_{kn} - 1)$.

3. For every fixed channel n, L_{kn} are *i.i.d.* random variables over different k.

Proof (Lemma 3.3)**.** 1. We have $L_{kn} = 1$ if, on the first slot of serving channel n, either a dummy packet is transmitted or a data packet is transmitted but the channel is OFF. This event occurs with probability

$$\left(1 - \frac{\mathsf{P}_{n,01}^{(M)}}{\omega_n(t)}\right) + \frac{\mathsf{P}_{n,01}^{(M)}}{\omega_n(t)} (1 - \omega_n(t)) = 1 - \mathsf{P}_{n,01}^{(M)}.$$

Next, $L_{kn} = j \geq 2$ if in the first slot a data packet is successfully served, and this is followed by $(j-2)$ consecutive ON slots and one OFF slot. This happens with probability $\mathsf{P}_{n,01}^{(M)} (\mathsf{P}_{n,11})^{(j-2)} \mathsf{P}_{n,10}$. The expectation of L_{kn} can be directly computed from the probability mass function.

2. We observe that one data packet is served in every slot of L_{kn} except for the last one (when a dummy packet is sent over channel n, we have $L_{kn} = 1$ and zero data packets are served).

3. At the beginning of every L_{kn}, we observe from the equivalent fictitious round robin policy that $\mathsf{RR(M)}$ effectively fixes $\mathsf{P}_{n,01}^{(M)}$ as the current information state, regardless of the true current state $\omega_n(t)$. Neglecting $\omega_n(t)$ is to discard all system history, including all past $L_{k'n}$ values for all $k' < k$. Thus L_{kn} are i.i.d.. Specifically, for any $k' < k$ and integers $l_{k'}$ and l_k, we have

$$\Pr\left[L_{kn} = l_k \mid L_{k'n} = l_{k'}\right] = \Pr\left[L_{kn} = l_k\right]. \quad \Box$$

Using Lemma 3.3, we derive the throughput vector rendered by $\mathsf{RR(M)}$. Fix an integer $K > 0$. By Lemma 3.3, the time average throughput over channel n after all channels finish their Kth rounds, which we denote by $\mu_n(K)$, is

$$\mu_n(K) \triangleq \frac{\sum_{k=1}^{K} (L_{kn} - 1)}{\sum_{k=1}^{K} \sum_{n=1}^{M} L_{kn}}.$$

Passing $K \to \infty$, we get

$$\begin{aligned}
\lim_{K\to\infty} \mu_n(K) &= \lim_{K\to\infty} \frac{\sum_{k=1}^{K}(L_{kn}-1)}{\sum_{k=1}^{K}\sum_{n=1}^{M} L_{kn}} \\
&= \lim_{K\to\infty} \frac{(1/K)\sum_{k=1}^{K}(L_{kn}-1)}{\sum_{n=1}^{M}(1/K)\sum_{k=1}^{K} L_{kn}} \\
&\overset{(a)}{=} \frac{\mathbb{E}[L_{1n}]-1}{\sum_{n=1}^{M}\mathbb{E}[L_{1n}]} \\
&\overset{(b)}{=} \frac{\mathsf{P}_{n,01}(1-(1-x_n)^M)/(x_n\mathsf{P}_{n,10})}{M+\sum_{n=1}^{M}\mathsf{P}_{n,01}(1-(1-x_n)^M)/(x_n\mathsf{P}_{n,10})},
\end{aligned} \tag{3.6}$$

where (a) follows the Law of Large Numbers (noting that L_{kn} are i.i.d. over k) and (b) uses (3.5).

3.2.2 Example of Symmetric Channels

We are particularly interested in the sum throughput under $\mathsf{RR}(\mathsf{M})$ when channels are symmetric. In this case, by channel symmetry, every channel has the same throughput. From (3.6), the sum throughput is

$$\sum_{n=1}^{M}\lim_{K\to\infty}\mu_n(K) = \frac{\mathsf{P}_{01}(1-(1-x)^M)}{x\,\mathsf{P}_{10}+\mathsf{P}_{01}(1-(1-x)^M)},$$

where, in the last term, the subscript n is dropped due to channel symmetry. It is handy to define a function $c_{(\cdot)}: \mathbb{N} \to \mathbb{R}$ as

$$c_M \triangleq \frac{\mathsf{P}_{01}(1-(1-x)^M)}{x\,\mathsf{P}_{10}+\mathsf{P}_{01}(1-(1-x)^M)}, \quad x \triangleq \mathsf{P}_{01}+\mathsf{P}_{10}, \tag{3.7}$$

and we define $c_\infty \triangleq \lim_{M\to\infty} c_M = \mathsf{P}_{01}/(x\mathsf{P}_{10}+\mathsf{P}_{01})$ (noting that $x<1$ because every channel is assumed positively correlated over slots). The function $c_{(\cdot)}$ will be used extensively in this chapter. We summarize the above analysis in the next lemma.

Lemma 3.4. Policy $\mathsf{RR(M)}$ serves channel $n \in \{1, 2, \dots, M\}$ with throughput

$$\frac{\mathsf{P}_{n,01}(1-(1-x_n)^M)/(x_n\mathsf{P}_{n,10})}{M+\sum_{n=1}^{M}\mathsf{P}_{n,01}(1-(1-x_n)^M)/(x_n\mathsf{P}_{n,10})}.$$

Particularly, in symmetric channels, the sum throughput under $\mathsf{RR(M)}$ is

$$c_M = \frac{\mathsf{P}_{01}(1-(1-x)^M)}{x\,\mathsf{P}_{10}+\mathsf{P}_{01}(1-(1-x)^M)}, \quad x = \mathsf{P}_{01}+\mathsf{P}_{10},$$

and every channel has throughput c_M/M.

We remark that the sum throughput c_M of $\mathsf{RR(M)}$ in the symmetric case is non-decreasing in M, and thus can be improved by serving more channels. When channel memory is neglected, Li and Neely [2010a] shows that the maximum sum throughput is equal to $\pi_{\mathsf{ON}} = c_1$, which is strictly less than the memory-assisted throughput c_M whenever $M \geq 2$ and $x < 1$. Interestingly, here we see that the sum throughput is improved by having multiuser diversity and channel memory in the network, even though instantaneous channel states are never known.

3.2.3 Asymptotical Throughput Optimality

Next, in symmetric channels, we quantify how close the sum throughput c_M is to optimal. The following lemma presents a useful upper bound on the maximum sum throughput.

Lemma 3.5 (Ahmad et al. [2009], Zhao et al. [2008])**.** In symmetric channels, any scheduling policy has sum throughput less than or equal to c_∞.[5]

[5] We note that the throughput analysis in Zhao et al. [2008] makes a minor assumption on the existence of some limiting time average. Using similar ideas of Zhao et al. [2008], in Theorem 3.12 of Section 3.3.2 we will construct an upper bound on the maximum sum throughput for general positively correlated Markov ON/OFF channels. When restricted to the symmetric case, we get the same upper bound without any assumption.

By Lemma 3.4 and 3.5, the loss of sum throughput of $\mathsf{RR(M)}$ is no larger than $c_\infty - c_M$. Define $\widetilde{c}_M$ as

$$\widetilde{c}_M \triangleq \frac{\mathsf{P}_{01}(1-(1-x)^M)}{x\mathsf{P}_{10}+\mathsf{P}_{01}} = c_\infty(1-(1-x)^M)$$

and note that $\widetilde{c}_M \leq c_M \leq c_\infty$. Then, the throughput loss of $\mathsf{RR(M)}$ is bounded by

$$c_\infty - c_M \leq c_\infty - \widetilde{c}_M = c_\infty(1-x)^M. \tag{3.8}$$

The last term of (3.8) decreases to zero geometrically fast as M increases. This indicates that $\mathsf{RR(M)}$ yields near-optimal sum throughput even when it only serves a moderately large number of channels.

3.3 Randomized Round Robin Policy RandRR

Lemma 3.4 specifies the throughput vector achieved by $\mathsf{RR(M)}$ over a particular collection of M channels. Generally, we are interested in the set of throughput vectors achieved by randomly mixing $\mathsf{RR(M)}$-like policies over different channel subsets and allowing a different round-robin ordering on each subset. First, we need to generalize the $\mathsf{RR(M)}$ policy. Let Φ denote the set of all N-dimensional binary vectors excluding the zero vector $(0, 0, \ldots, 0)$. For any binary vector $\boldsymbol{\phi} = (\phi_n)_{n=1}^N$ in Φ, we say channel n is *active* in $\boldsymbol{\phi}$ if $\phi_n = 1$. Each vector $\boldsymbol{\phi} \in \Phi$ represents a different subset of active channels. We denote by $M(\boldsymbol{\phi})$ the number of active channels in $\boldsymbol{\phi}$.

For each $\boldsymbol{\phi} \in \Phi$, consider the following round robin policy $\mathsf{RR}(\boldsymbol{\phi})$ that serves active channels in $\boldsymbol{\phi}$ in every round.

Dynamic Round Robin Policy $\mathsf{RR}(\boldsymbol{\phi})$:

1. Deciding the service order in each round:

At the beginning of each round, we denote by τ_n the time duration between the last use of channel n and the beginning of the current round. Active channels in $\boldsymbol{\phi}$ are served in the decreasing order of τ_n in this round (in other words, the active channel that is *least recently used* is served first).

2. On each active channel in a round:

   (a) Suppose at time t the base station switches to channel n. Then, it transmits a *data* packet to user n with probability $\mathsf{P}_{n,01}^{(M(\boldsymbol{\phi}))}/\omega_n(t)$ and a *dummy* packet otherwise. In both cases, ACK/NACK information is received at the end of the slot.

   (b) At time $(t+1)$, if a dummy packet is sent at time t, the base station switches to the next active channel following the order given in Step 1. Otherwise, it keeps transmitting data packets over channel n until a NACK is received; then, it switches to the next active channel and go to Step 2a.

3. Update $\boldsymbol{\omega}(t)$ according to (3.1) in every slot.

Using $\mathsf{RR}(\boldsymbol{\phi})$ as a building block, we consider the following class of *randomized round robin policies.*

Randomized Round Robin Policy RandRR:

1. Pick $\boldsymbol{\phi} \in \Phi$ with probability $\alpha_{\boldsymbol{\phi}}$, where $\sum_{\boldsymbol{\phi}\in\Phi} \alpha_{\boldsymbol{\phi}} = 1$.

2. Run policy $\mathsf{RR}(\boldsymbol{\phi})$ for one round. Then go to Step 1.

Note that the RandRR policy may serve active channels in different order in different rounds, according to the least-recently-used service order. This allows more time for OFF channels to return to better information states (note that $\mathsf{P}_{n,01}^{(k)}$ is nondecreasing in k) and improves throughput. The next lemma guarantees the feasibility of executing a $\mathsf{RR}(\boldsymbol{\phi})$ policy in every round of RandRR. In particular, similar to Lemma 3.2, whenever the base station switches to a new channel n, we need $\omega_n(t) \geq \mathsf{P}_{n,01}^{(M(\boldsymbol{\phi}))}$ in Step 2a of $\mathsf{RR}(\boldsymbol{\phi})$.

Lemma 3.6 (Proof in Section 3.8.2)**.** When $\mathsf{RR}(\boldsymbol{\phi})$ is chosen by RandRR for a new round of transmission, every active channel n in $\boldsymbol{\phi}$ starts with an information state no worse than $\mathsf{P}_{n,01}^{(M(\boldsymbol{\phi}))}$.

Although RandRR randomly selects a subset of channels in every round and serves them in an order that depends on previous choices, we can surprisingly analyze its throughput. This is done by using the throughput analysis of $\mathsf{RR}(\mathsf{M})$, shown in the following corollary of Lemma 3.3.

Corollary 3.7. For each policy $\mathsf{RR}(\boldsymbol{\phi})$, $\boldsymbol{\phi} \in \Phi$, within time periods in which $\mathsf{RR}(\boldsymbol{\phi})$ is executed by RandRR, we denote by $L_{kn}^{\boldsymbol{\phi}}$ the duration of the kth time the base station stays with active channel n. Then:

1. The probability mass function of $L_{kn}^{\boldsymbol{\phi}}$ is independent of k, and is

$$L_{kn}^{\boldsymbol{\phi}} = \begin{cases} 1 & \text{with prob. } 1 - \mathsf{P}_{n,01}^{(M(\boldsymbol{\phi}))} \\ j \geq 2 & \text{with prob. } \mathsf{P}_{n,01}^{(M(\boldsymbol{\phi}))} \left(\mathsf{P}_{n,11}\right)^{(j-2)} \mathsf{P}_{n,10}. \end{cases}$$

   As a result, for all $k \in \mathbb{N}$ we have

$$\mathbb{E}\left[L_{kn}^{\boldsymbol{\phi}}\right] = 1 + \frac{\mathsf{P}_{n,01}^{(M(\boldsymbol{\phi}))}}{\mathsf{P}_{n,10}}. \tag{3.9}$$

2. The number of data packets served in $L_{kn}^{\boldsymbol{\phi}}$ is $(L_{kn}^{\boldsymbol{\phi}} - 1)$.

3. For every fixed $\boldsymbol{\phi}$ and every fixed active channel n in $\boldsymbol{\phi}$, the time durations $L_{kn}^{\boldsymbol{\phi}}$ are i.i.d. random variables over different k.

3.3.1 Achievable Network Capacity: An Inner Bound

Using Corollary 3.7, we present the throughput region achieved by the class of RandRR policies; this is our inner bound on the network capacity region. For each $\mathsf{RR}(\boldsymbol{\phi})$ policy, we define an N-dimensional vector $\boldsymbol{\eta}^{\boldsymbol{\phi}} = (\eta_n^{\boldsymbol{\phi}})_{n=1}^N$ where

$$\eta_n^{\boldsymbol{\phi}} \triangleq \begin{cases} \frac{\mathbb{E}\left[L_{1n}^{\boldsymbol{\phi}}\right]-1}{\sum_{n:\phi_n=1}\mathbb{E}\left[L_{1n}^{\boldsymbol{\phi}}\right]} & \text{if channel } n \text{ is active in } \boldsymbol{\phi}, \\ 0 & \text{otherwise,} \end{cases} \tag{3.10}$$

where $\mathbb{E}\left[L_{1n}^{\boldsymbol{\phi}}\right]$ is given in (3.9). Intuitively, by the analysis prior to Lemma 3.4, policy $\mathsf{RR}(\boldsymbol{\phi})$ yields throughput $\eta_n^{\boldsymbol{\phi}}$ over channel $n \in \{1, 2, \ldots, N\}$. Incorporating all possible random mixtures of $\mathsf{RR}(\boldsymbol{\phi})$ policies for different channel subsets $\boldsymbol{\phi}$, RandRR shall support any data rate vector that is entrywise dominated by a convex combination of vectors $\{\boldsymbol{\eta}^{\boldsymbol{\phi}}\}_{\boldsymbol{\phi}\in\Phi}$.

Theorem 3.8 (Generalized Inner Capacity Bound; proof in Section 3.8.3)**.** The class of RandRR policies supports all data rate vectors $\boldsymbol{\lambda}$ in the set Λ_{int} defined as

$$\Lambda_{\text{int}} \triangleq \left\{ \boldsymbol{\lambda} \mid \mathbf{0} \leq \boldsymbol{\lambda} \leq \boldsymbol{\mu},\ \boldsymbol{\mu} \in \operatorname{conv}\left(\left\{\boldsymbol{\eta}^{\boldsymbol{\phi}}\right\}_{\boldsymbol{\phi}\in\Phi}\right)\right\},$$

where $\boldsymbol{\eta}^{\boldsymbol{\phi}}$ is defined in (3.10), $\operatorname{conv}(A)$ denotes the convex hull of set A, and $\leq$ is taken entrywise.

Applying Theorem 3.8 to symmetric channels yields the following corollary.

Corollary 3.9 (Inner Capacity Bound for Symmetric Channels)**.** In symmetric channels, the class of RandRR policies supports all rate vectors $\boldsymbol{\lambda} \in \Lambda_{\text{int}}$ where

$$\Lambda_{\text{int}} = \left\{ \boldsymbol{\lambda} \mid \mathbf{0} \leq \boldsymbol{\lambda} \leq \boldsymbol{\mu}, \boldsymbol{\mu} \in \operatorname{conv}\left(\left\{\frac{c_{M(\boldsymbol{\phi})}}{M(\boldsymbol{\phi})}\boldsymbol{\phi}\right\}_{\boldsymbol{\phi}\in\Phi}\right)\right\},$$

where $c_{M(\boldsymbol{\phi})}$ is defined in (3.7).

The inner capacity bound Λ_{int} in Theorem 3.8 comprises all rate vectors that can be written as a convex combination of the zero vector and all throughput vectors $\boldsymbol{\eta}^{\phi}$ (see (3.10)) yielded by round robin policies $\mathsf{RR}(\boldsymbol{\phi})$ that serve different subsets $\boldsymbol{\phi}$ of channels. This convex hull characterization shows that the inner bound Λ_{int} contains a large class of policies and is intuitively near optimal because it is constructed by randomizing the efficient round robin policies $\mathsf{RR}(\mathsf{M})$ over all subsets of channels. A simple example of the inner bound Λ_{int} is provided later in Section 3.3.3.

We also remark that, although Λ_{int} is conceptually simple, supporting any given rate vector $\boldsymbol{\lambda}$ within Λ_{int} could be difficult because finding the right convex combination of round robin policies $\mathsf{RR}(\boldsymbol{\phi})$ that supports $\boldsymbol{\lambda}$ is of exponential complexity. In Section 3.5, we provide a simple queue-dependent round robin policy that supports all data rate vectors within the inner bound Λ_{int} with polynomial complexity.

3.3.2 Outer Capacity Bound

To study the tightness of the inner capacity region we create in the previous section, we construct an outer bound on the network capacity region Λ using several novel ideas. First, by state aggregation, we transform the information state process $\{\omega_n(t)\}$ for each channel n into non-stationary two-state Markov chains (in Fig. 3.4 later). Second, we create a set of bounding stationary Markov chains (in Fig. 3.5 later), which has the structure of a multi-armed bandit system. Finally, we create an outer capacity bound by relating the bounding model to the original non-stationary Markov chains using stochastic coupling. We note that, since the control of the information state processes $\{\omega_n(t)\}$ over all channels n is a restless multi-armed bandit problem, it is interesting to see how we bound the optimal performance of a restless bandit by a related multi-armed bandit system.

We first *map* channel information states $\omega_n(t)$ into *modes* for each $n \in \{1, 2, \dots, N\}$. Inspired by (3.3), we observe that each channel n must be in one of the following two modes:

M1 The last observed state is ON, and the channel has not been seen (through feedback) to turn OFF. In this mode the information state $\omega_n(t) \in [\pi_{n,\mathsf{ON}}, \mathsf{P}_{n,11}]$.

M2 The last observed state is OFF, and the channel has not been seen to turned ON. Here $\omega_n(t) \in [\mathsf{P}_{n,01}, \pi_{n,\mathsf{ON}}]$.

On channel n, we recall that $\mathcal{W}_n = \{\mathsf{P}_{n,01}^{(k)}, \mathsf{P}_{n,11}^{(k)} : k \in \mathbb{N}\} \cup \{\pi_{n,\mathsf{ON}}\}$ is the state space of $\omega_n(t)$. We define a map $f_n : \mathcal{W}_n \to \{\mathsf{M1}, \mathsf{M2}\}$ where

$$f_n(\omega_n(t)) = \begin{cases} \mathsf{M1} & \text{if } \omega_n(t) \in (\pi_{n,\mathsf{ON}}, \mathsf{P}_{n,11}], \\ \mathsf{M2} & \text{if } \omega_n(t) \in [\mathsf{P}_{n,01}, \pi_{n,\mathsf{ON}}]. \end{cases}$$

This mapping is illustrated in Fig. 3.3.

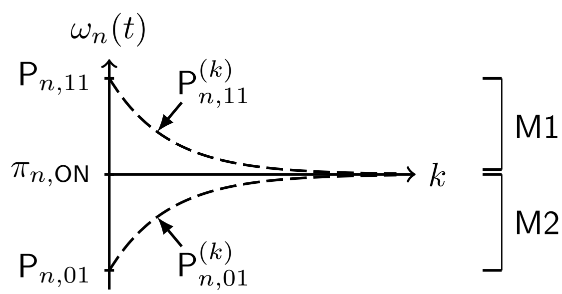

Figure 3.3: The mapping f_n from information states $\omega_n(t)$ to modes $\{\mathsf{M1}, \mathsf{M2}\}$.

For any information state process $\{\omega_n(t)\}_{t\in\mathbb{Z}^+}$ (controlled by some scheduling policy), the corresponding mode transition process under f_n can be represented by the Markov chains shown in Fig. 3.4. Specifically, when channel n is served in a slot, the associated mode transition follows the upper non-stationary chain of Fig. 3.4. When channel n is idled in a slot, the mode transition follows the lower stationary chain of Fig. 3.4. In the upper chain of Fig. 3.4, regardless what the current mode is, mode M1 is visited in the next slot if and only if channel n is ON in the current slot, which occurs with probability $\omega_n(t)$. In the lower chain of Fig. 3.4, when channel n is idled, its information state changes from a k-step transition probability to a $(k+1)$-step transition probability with the same most recent observed channel state. Therefore, the next mode stays the same as the current mode. We emphasize that, in the upper chain of Fig. 3.4, at mode M1 we always have $\omega_n(t) \leq \mathsf{P}_{n,11}$, and at mode M2 it is always $\omega_n(t) \leq \pi_{n,\mathsf{ON}}$. A packet is served if and only if M1 is visited in the upper chain of Fig. 3.4.

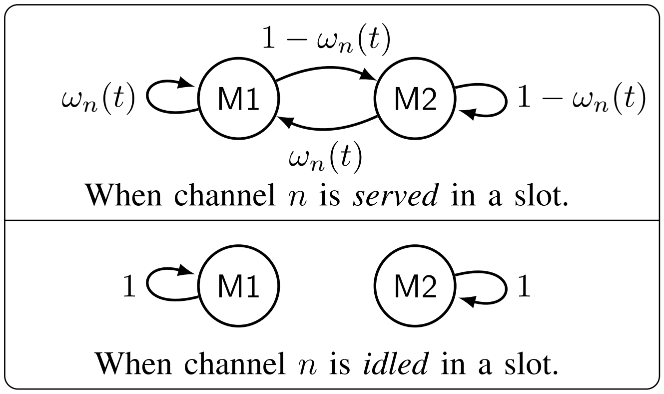

Figure 3.4: Mode transition diagrams for the real channel n.

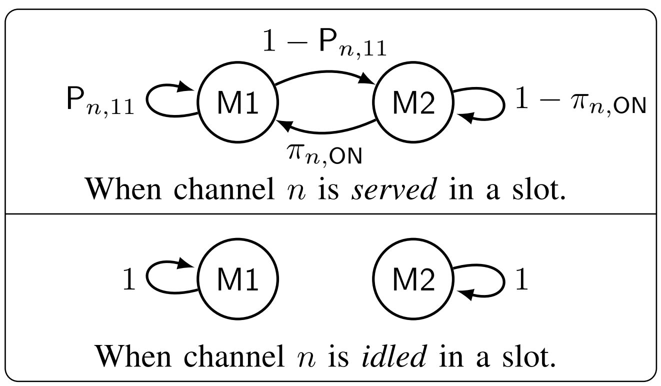

Figure 3.5: Mode transition diagrams for the fictitious channel n.

To upper bound throughput, we compare Fig. 3.4 to the mode transition diagrams in Fig. 3.5 that correspond to a fictitious model for channel n. This fictitious channel has constant information state $\omega_n(t) = \mathsf{P}_{n,11}$ whenever it is in mode M1, and $\omega_n(t) = \pi_{n,\mathsf{ON}}$ whenever it is in M2. In other words, when the fictitious channel n is in either mode M1 or M2, it sets its current information state to be the best state possible when the corresponding real channel n is in the same mode. It follows that, when both the real channel n and the fictitious channel n are served, the probabilities of the transitions $\mathsf{M1} \to \mathsf{M1}$ and $\mathsf{M2} \to \mathsf{M1}$ in the upper chain of Fig. 3.5 are greater than or equal to those in Fig. 3.4, respectively. In other words, the upper chain of Fig. 3.5 is *more likely* to go to mode $\mathsf{M1}$ and serve packets than that of Fig. 3.4. Therefore, intuitively, if we serve both the real channel n and the fictitious channel n in the same infinite sequence of time slots, the fictitious channel n yields higher throughput than the real channel n. This observation is made precise by the next lemma.

Lemma 3.10 (Proof in Section 3.8.5)**.** Consider two discrete-time Markov chains $\{X(t)\}$ and $\{Y(t)\}$ both with state space $\{0, 1\}$. Suppose $\{X(t)\}$ is stationary and ergodic with transition probability matrix

$$\mathbf{P} = \begin{bmatrix} \mathsf{P}_{00} & \mathsf{P}_{01} \\ \mathsf{P}_{10} & \mathsf{P}_{11} \end{bmatrix},$$

and $\{Y(t)\}$ is non-stationary with

$$\mathbf{Q}(t) = \begin{bmatrix} \mathsf{Q}_{00}(t) & \mathsf{Q}_{01}(t) \\ \mathsf{Q}_{10}(t) & \mathsf{Q}_{11}(t) \end{bmatrix}.$$

Assume $\mathsf{P}_{01} \geq \mathsf{Q}_{01}(t)$ and $\mathsf{P}_{11} \geq \mathsf{Q}_{11}(t)$ for all t. In $\{X(t)\}$, let $\pi_X(1)$ denote the stationary probability of state 1; $\pi_X(1) = \mathsf{P}_{01}/(\mathsf{P}_{01} + \mathsf{P}_{10})$. In $\{Y(t)\}$, define

$$\pi_Y(1) \triangleq \limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} Y(t)$$

as the limiting fraction of time $\{Y(t)\}$ stays at state 1. Then, we have $\pi_X(1) \geq \pi_Y(1)$.

We note that executing a scheduling policy in the network is to generate a sequence of channel selection decisions. By Lemma 3.10, if we apply the same sequence of channel selection decisions of some scheduling policy to the set of fictitious channels, we get higher throughput on every channel. A direct consequence of this is that the maximum sum throughput over the fictitious channels is greater than or equal to that over the real channels.

Lemma 3.11. The maximum sum throughput over the set of fictitious channels is no more than

$$\max_{n\in\{1,2,\dots,N\}}\{c_{n,\infty}\}, \quad c_{n,\infty} \triangleq \frac{\mathsf{P}_{n,01}}{x_n\mathsf{P}_{n,10}+\mathsf{P}_{n,01}}.$$

Proof (Lemma 3.11)**.** We note that finding the maximum sum throughput over fictitious channels in Fig. 3.5 is equivalent to solving a multi-armed bandit problem [Gittins 1989] with each channel acting as an arm (noting in Fig. 3.5 that a channel can change mode only when it is served), and one unit of reward is earned if a packet is delivered (recalling that a packet is served if and only if mode $\mathsf{M1}$ is visited in the upper chain of Fig. 3.5). The optimal solution to the multi-armed bandit system is to always play the arm (channel) with the largest average reward (throughput). The average reward over channel n is equal to the stationary probability of mode $\mathsf{M1}$ in the upper chain of Fig. 3.5, which is

$$\frac{\pi_{n,\mathsf{ON}}}{\mathsf{P}_{n,10}+\pi_{n,\mathsf{ON}}} = \frac{\mathsf{P}_{n,01}}{x_n\mathsf{P}_{n,10}+\mathsf{P}_{n,01}}.$$

This finishes the proof. □

Together with the fact that throughput over any real channel n cannot exceed its stationary ON probability $\pi_{n,\mathsf{ON}}$, we have constructed an outer bound on the network capacity region Λ, formalized in the next theorem (the proof follows the above discussions and is omitted).

Theorem 3.12 (Generalized Outer Capacity Bound)**.** Every supportable throughput vector $\boldsymbol{\lambda} = (\lambda_n)_{n=1}^N$ necessarily satisfies

$$\lambda_n \le \pi_{n,\mathsf{ON}}, \quad \text{for all } n \in \{1,2,\dots,N\},$$

$$\sum_{n=1}^N \lambda_n \le \max_{n\in\{1,2,\dots,N\}}\{c_{n,\infty}\} = \max_{n\in\{1,2,\dots,N\}}\left\{\frac{\mathsf{P}_{n,01}}{x_n\mathsf{P}_{n,10}+\mathsf{P}_{n,01}}\right\}.$$

These $(N+1)$ hyperplanes create an outer capacity bound Λ_{out} on Λ.

Corollary 3.13 (Outer Capacity Bound for Symmetric Channels)**.** In symmetric channels with $\mathsf{P}_n = \mathsf{P}$, $c_{n,\infty} = c_\infty$, and $\pi_{n,\mathsf{ON}} = \pi_{\mathsf{ON}}$ for all n, we have

$$\Lambda_{\text{out}} = \left\{ \boldsymbol{\lambda} \geq \mathbf{0} \mid \sum_{n=1}^{N} \lambda_n \leq c_\infty,\ \lambda_n \leq \pi_{\mathsf{ON}} \text{ for } 1 \leq n \leq N \right\}, \tag{3.11}$$

where $\geq$ is taken entrywise.

We note that Lemma 3.5 in Section 3.2.2 directly follows Corollary 3.13.

3.3.3 Example of Symmetric Channels

Here we consider a two-user example on symmetric channels. For simplicity, we drop the subscript n in notations. From Corollary 3.13, we have the outer bound

$$\Lambda_{\text{out}} = \left\{ (\lambda_1, \lambda_2) \;\middle|\; \begin{array}{c} 0 \leq \lambda_n \leq \mathsf{P}_{01}/x, \text{ for } 1 \leq n \leq 2, \\ \lambda_1 + \lambda_2 \leq \mathsf{P}_{01}/(x\mathsf{P}_{10} + \mathsf{P}_{01}), \\ x = \mathsf{P}_{01} + \mathsf{P}_{10} \end{array} \right\}.$$

For the inner bound Λ_{int}, policy RandRR can execute three round robin policies $\mathsf{RR}(\boldsymbol{\phi})$ for $\boldsymbol{\phi} \in \Phi = \{(1,1), (0,1), (1,0)\}$. From Corollary 3.9, we have

$$\Lambda_{\text{int}} = \left\{ (\lambda_1, \lambda_2) \;\middle|\; \begin{array}{c} 0 \leq \lambda_n \leq \mu_n, \text{ for } 1 \leq n \leq 2, \\ (\mu_1, \mu_2) \in \operatorname{conv}\left(\left\{ (\frac{c_2}{2}, \frac{c_2}{2}), (c_1, 0), (0, c_1) \right\}\right) \end{array} \right\}.$$

In the case of $\mathsf{P}_{01} = \mathsf{P}_{10} = 0.2$, the two bounds λ_{int} and Λ_{out} are shown in Fig. 3.6.

In Fig. 3.6, we also compare Λ_{int} and Λ_{out} with other rate regions. Set Λ_{ideal} is the ideal capacity region when instantaneous channel states are known without causing any (timing) overhead [Tassiulas and Ephremides 1993]. Zhao et al. [2008] shows that the maximum sum throughput in this network is achieved at point $A = (0.325, 0.325)$. The (unknown) network capacity region Λ is bounded between Λ_{int} and Λ_{out}, and has

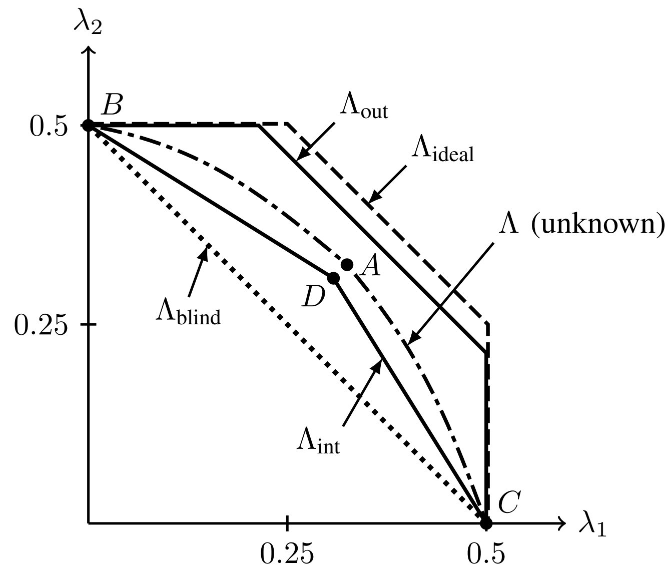

Figure 3.6: Comparison of throughput regions under different assumptions.

boundary points B, A, and C. Since the boundary of Λ is a concave curve connecting B, A, and C, we envision that Λ shall contain but be very close to Λ_{int}.

The throughput region Λ_{blind} in Fig. 3.6 is rendered by neglecting channel memory and treating channels as i.i.d. over slots [Li and Neely 2010a]. The throughput gain $\Lambda_{\text{int}} \setminus \Lambda_{\text{blind}}$, as much as 23% in this example, is achieved by using channel memory. In general, if channels are symmetric and treated as i.i.d. over slots, the maximum sum throughput is $\pi_{\mathsf{ON}} = c_1$. The maximum throughput gain of RandRR that uses channel memory is $c_N - c_1$, which, as $N \to \infty$, converges to

$$c_\infty - c_1 = \frac{\mathsf{P}_{01}}{x\mathsf{P}_{10} + \mathsf{P}_{01}} - \frac{\mathsf{P}_{01}}{\mathsf{P}_{01} + \mathsf{P}_{10}},$$

which is controlled by the factor $x = \mathsf{P}_{01} + \mathsf{P}_{10}$.

3.3.4 A Heuristically Tighter Inner Bound

Ahmad et al. [2009] shows that the following policy maximizes the sum throughput in the special case of symmetric channels:

Serve channels in a fixed circular order, where on each channel keep transmitting data packets until a NACK is received.

In the above two-user example, this policy achieves throughput vector A in Fig. 3.7. If we

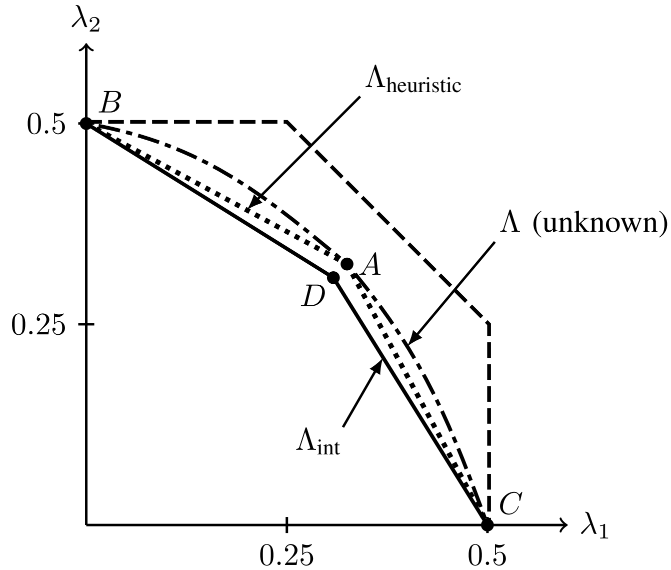

Figure 3.7: Comparison of our inner capacity bound Λ_{int}, the unknown network capacity region Λ, and a heuristically better inner capacity bound $\Lambda_{\text{heuristic}}$.

replace our round robin policy $\mathsf{RR}(\phi)$ by this one, heuristically we are able to construct a tighter inner capacity bound. For example, we can support the tighter inner bound $\Lambda_{\text{heuristic}}$ in Fig. 3.7 by appropriate time sharing of the above sum-throughput-optimal policy applied to different subsets of channels. However, we note that this approach is difficult to analyze because the $\{L_{kn}\}$ process (see (3.4)) forms a high-order Markov chain. Our inner bound Λ_{int} provides a good throughput guarantee for this class of heuristic policies.

3.4 Tightness of Inner Capacity Bound: Symmetric Case

Next we bound the closeness of the boundaries of Λ_{int} and Λ in the case of symmetric channels. In Section 3.2.2, by choosing $M = N$, we have provided such analysis for the boundary point in the direction $(1, 1, \ldots, 1)$. Here we generalize to all boundary points. Define

$$\mathcal{V} \triangleq \left\{ (v_n)_{n=1}^{N} \;\middle|\; \begin{array}{c} v_n \geq 0 \text{ for } 1 \leq n \leq N, \\ v_n > 0 \text{ for at least one } n \end{array} \right\}$$

as a set of directional vectors. For any $\boldsymbol{v} \in \mathcal{V}$, let $\boldsymbol{\lambda}^{\text{int}} = (\lambda_n^{\text{int}})_{n=1}^{N}$ and $\boldsymbol{\lambda}^{\text{out}} = (\lambda_1^{\text{out}})_{n=1}^{N}$ be the boundary point of Λ_{int} and Λ_{out} in the direction of $\boldsymbol{v}$, respectively. It is useful to compute $\sum_{n=1}^{N}(\lambda_n^{\text{out}} - \lambda_n^{\text{int}})$, because it upper bounds the loss of the sum throughput of Λ_{int} from Λ in the direction of $\boldsymbol{v}$.[6] Computing $\boldsymbol{\lambda}^{\text{int}}$ in an arbitrary direction is difficult. Thus we find an upper bound on $\sum_{n=1}^{N}(\lambda_n^{\text{out}} - \lambda_n^{\text{int}})$.

3.4.1 Preliminary

To have more intuitions on Λ_{int}, we consider an example of $N = 3$ users. We are interested in the boundary point of Λ_{int} in the direction of $\boldsymbol{v} = (1, 2, 1)$. Consider two RandRR policies ψ_1 and ψ_2 defined as follows.

$$\text{For } \psi_1, \text{ choose } \begin{cases} \boldsymbol{\phi}^1 = (1,0,0) & \text{with prob. } 1/4 \\ \boldsymbol{\phi}^2 = (0,1,0) & \text{with prob. } 1/2 \\ \boldsymbol{\phi}^3 = (0,0,1) & \text{with prob. } 1/4 \end{cases}$$

$$\text{For } \psi_2, \text{ choose } \begin{cases} \boldsymbol{\phi}^4 = (1,1,0) & \text{with prob. } 1/2 \\ \boldsymbol{\phi}^5 = (0,1,1) & \text{with prob. } 1/2 \end{cases}$$

[6] The sum $\sum_{n=1}^{N}(\lambda_n^{\text{out}} - \lambda_n^{\text{int}})$ also bounds the closeness between Λ_{out} and Λ.

Both ψ_1 and ψ_2 support data rate vectors in the direction of $(1, 2, 1)$. However, using the analysis for Lemma 3.4 and Theorem 3.8, we know ψ_1 supports throughput vector

$$\frac{1}{4}\begin{bmatrix} c_1 \\ 0 \\ 0 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 0 \\ c_1 \\ 0 \end{bmatrix} + \frac{1}{4}\begin{bmatrix} 0 \\ 0 \\ c_1 \end{bmatrix} = \frac{c_1}{4}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix},$$

while ψ_2 supports

$$\frac{1}{2}\begin{bmatrix} c_2/2 \\ c_2/2 \\ 0 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 0 \\ c_2/2 \\ c_2/2 \end{bmatrix} = \frac{c_2}{4}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \geq \frac{c_1}{4}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix},$$

where c_1 and c_2 are defined in (3.7). We see that ψ_2 achieves a rate vector closer to the boundary of Λ_{int} than ψ_1 does. It is because every sub-policy of ψ_2, namely $\mathsf{RR}(\boldsymbol{\phi}^4)$ and $\mathsf{RR}(\boldsymbol{\phi}^5)$, supports sum throughput c_2 (by Lemma 3.4), where those of ψ_1 only support c_1. In other words, policy ψ_2 has better *multiuser diversity gain* than ψ_1 does. This example suggests that we can find a good lower bound on $\boldsymbol{\lambda}^{\text{int}}$ by exploring to what extent the multiuser diversity can be exploited. We start with the following definition.

Definition 3.14. For any $\boldsymbol{v} \in \mathcal{V}$, we say $\boldsymbol{v}$ is *d-user diverse* if $\boldsymbol{v}$ can be written as a *positive* combination of vectors in Φ_d, where Φ_d denotes the set of N-dimensional binary vectors having d entries be 1. Define

$$d(\boldsymbol{v}) \triangleq \max_{1 \leq d \leq N} \{d \mid \boldsymbol{v} \text{ is } d\text{-user diverse}\},$$

and we shall say $\boldsymbol{v}$ is *maximally $d(\boldsymbol{v})$-user diverse.*

The notion of $d(\boldsymbol{v})$ is well-defined because every $\boldsymbol{v}$ must be 1-user diverse.[7] Definition 3.14 is the most useful to us through the next lemma.

[7]The set $\Phi_1 = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_N\}$ is the collection of unit coordinate vectors where $\boldsymbol{e}_n$ has its nth entry be 1 and 0 otherwise. Any vector $\boldsymbol{v} \in \mathcal{V}$, $\boldsymbol{v} = (v_1, v_2, \ldots, v_N)$, can be written as $\boldsymbol{v} = \sum_{v_n > 0} v_n \boldsymbol{e}_n$.

Lemma 3.15 (Proof in Section 3.8.7)**.** The boundary point of Λ_{int} in the direction of $\boldsymbol{v} \in \mathcal{V}$ has sum throughput at least $c_{d(\boldsymbol{v})}$, where

$$c_{d(\boldsymbol{v})} \triangleq \frac{\mathsf{P}_{01}(1-(1-x)^{d(\boldsymbol{v})})}{x\,\mathsf{P}_{10}+\mathsf{P}_{01}(1-(1-x)^{d(\boldsymbol{v})})}, \quad x \triangleq \mathsf{P}_{01}+\mathsf{P}_{10}.$$

We provide a sketch of proof here. If direction $\boldsymbol{v}$ can be written as a positive weighted sum of vectors in $\Phi_{d(\boldsymbol{v})}$, we can normalize the weights and use them as probabilities to randomly mix $\mathsf{RR}(\boldsymbol{\phi})$ policies for all $\boldsymbol{\phi} \in \Phi_{d(\boldsymbol{v})}$. In this way, we achieve sum throughput $c_{d(\boldsymbol{v})}$ in every transmission round, and overall the throughput vector will be in the direction of $\boldsymbol{v}$.

Fig. 3.8 provides an example of Lemma 3.15 in the two-user symmetric system in Section 3.3.3. We observe that direction $(1, 1)$, the one that passes point D in Fig. 3.8, is

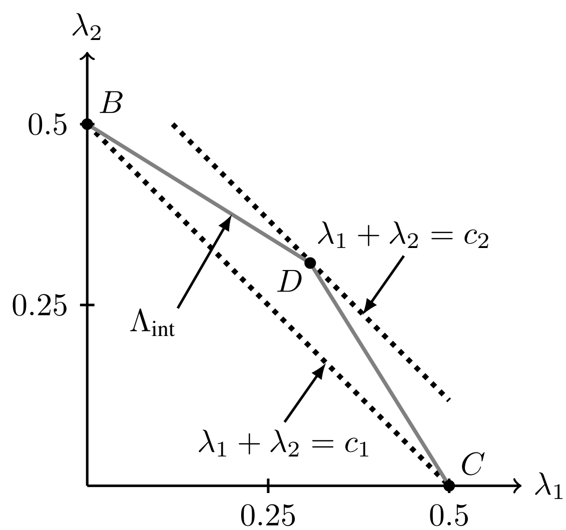

Figure 3.8: An example for Lemma 3.15 in the two-user symmetric network. Point B and C achieve sum throughput $c_1 = \pi_{\mathsf{ON}} = 0.5$, and the sum throughput at D is $c_2 \approx 0.615$. Any other boundary point of Λ_{int} has sum throughput between c_1 and c_2.

the only direction that is maximally 2-user diverse. The sum throughput c_2 is achieved at D. For all the other directions, they are maximally 1-user diverse and, from Fig. 3.8, only sum throughput c_1 is guaranteed along those directions. In general, geometrically we can show that a maximally d-user diverse vector, say $\boldsymbol{v}_d$, forms a smaller angle with the all-1 vector $(1, 1, \ldots, 1)$ than a maximally d'-user diverse vector, say $\boldsymbol{v}_{d'}$, does if $d' < d$. In other words, data rates along $\boldsymbol{v}_d$ are *more balanced* than those along $\boldsymbol{v}_{d'}$. Lemma 3.15 states that we guarantee to support higher sum throughput if the user traffic is more balanced.

3.4.2 Analysis

We use the notion of $d(\boldsymbol{v})$ to upper bound $\sum_{n=1}^{N}(\lambda_n^{\text{out}} - \lambda_n^{\text{int}})$ in any direction $\boldsymbol{v} \in \mathcal{V}$. Let $\boldsymbol{\lambda}^{\text{out}} = \theta \boldsymbol{\lambda}^{\text{int}}$ (i.e., $\lambda_n^{\text{out}} = \theta \lambda_n^{\text{int}}$ for all n) for some $\theta \geq 1$. By (3.11), the boundary of Λ_{out} is characterized by the interaction of the $(N+1)$ hyperplanes $\sum_{n=1}^{N} \lambda_n = c_\infty$ and $\lambda_n = \pi_{\mathsf{ON}}$ for each $n \in \{1, 2, \ldots, N\}$. Specifically, in any given direction, if we consider the cross points on all the hyperplanes in that direction, the boundary point $\boldsymbol{\lambda}^{\text{out}}$ is the one closest to the origin. We do not know which hyperplane $\boldsymbol{\lambda}^{\text{out}}$ is on, and thus need to consider all $(N+1)$ cases. If $\boldsymbol{\lambda}^{\text{out}}$ is on the plane $\sum_{n=1}^{N} \lambda_n = c_\infty$, i.e., $\sum_{n=1}^{N} \lambda_n^{\text{out}} = c_\infty$, we get

$$\sum_{n=1}^{N}(\lambda_n^{\text{out}} - \lambda_n^{\text{int}}) \overset{(a)}{\leq} c_\infty - c_{d(\boldsymbol{v})} \overset{(b)}{\leq} c_\infty (1-x)^{d(\boldsymbol{v})},$$

where (a) is by Lemma 3.15 and (b) is by (3.8). If $\boldsymbol{\lambda}^{\text{out}}$ is on the plane $\lambda_n = \pi_{\mathsf{ON}}$ for some n, then $\theta = \pi_{\mathsf{ON}}/\lambda_n^{\text{int}}$. It follows

$$\sum_{n=1}^{N}(\lambda_n^{\text{out}} - \lambda_n^{\text{int}}) = (\theta - 1)\sum_{n=1}^{N} \lambda_n^{\text{int}} \leq \left(\frac{\pi_{\mathsf{ON}}}{\lambda_n^{\text{int}}} - 1\right) c_\infty.$$

The above discussions lead to the next lemma.

Lemma 3.16. The loss of the sum throughput of Λ_{int} from Λ in the direction of $\boldsymbol{v}$ is upper bounded by

$$\begin{aligned} &\min\left[c_\infty (1-x)^{d(\boldsymbol{v})}, \min_{1 \leq n \leq N}\left\{\left(\frac{\pi_{\mathsf{ON}}}{\lambda_n^{\text{int}}} - 1\right) c_\infty\right\}\right] \\ &\quad = c_\infty \min\left[(1-x)^{d(\boldsymbol{v})}, \frac{\pi_{\mathsf{ON}}}{\max_{1\leq n \leq N}\{\lambda_n^{\text{int}}\}} - 1\right]. \end{aligned} \tag{3.12}$$

Lemma 3.16 shows that, if data rates of different users are more balanced, namely, have a larger $d(\boldsymbol{v})$, the loss of sum throughput is dominated by the first term in the minimum of (3.12), and decreases to 0 geometrically fast with $d(\boldsymbol{v})$. If data rates are biased toward a particular user, the second term in the minimum of (3.12) captures the throughput loss, which goes to 0 as the rate of the favored user goes to the single-user capacity π_{ON}.

3.5 Queue-Dependent Round Robin Policy QRR

Over positively correlated Markov ON/OFF channels that are possibly asymmetric, we present a queue-dependent dynamic round robin policy that supports any throughput vector within the inner throughput region Λ_{int} that is presented in Theorem 3.8. First, we set up additional notations. Let $a_n(t)$, for $1 \leq n \leq N$, be the number of exogenous packet arrivals destined for user n in slot t. Suppose $a_n(t)$ are independent across users, i.i.d. over slots with mean $\mathbb{E}[a_n(t)] = \lambda_n$, and $a_n(t)$ is bounded with $0 \leq a_n(t) \leq A_{\max}$, where $A_{\max}$ is a finite integer. Let $Q_n(t)$ be the backlog of user-n packets queued at the base station at time t. Define $\boldsymbol{Q}(t) \triangleq (Q_n(t))_{n=1}^N$ and suppose $Q_n(0) = 0$ for all n. The queue process $\{Q_n(t)\}$ evolves as

$$Q_n(t+1) = \max\left[Q_n(t) - \mu_n(s_n(t), t), 0\right] + a_n(t), \tag{3.13}$$

where $\mu_n(s_n(t), t) \in \{0, 1\}$ is the service rate allocated to user n in slot t. We have $\mu_n(s_n(t), t) = 1$ if user n is served and $s_n(t) = \mathsf{ON}$, and 0 otherwise. In the rest of the chapter, we drop $s_n(t)$ in $\mu_n(s_n(t), t)$ and use $\mu_n(t)$ for notational simplicity. We say the network is (strongly) stable if

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{n=1}^{N} \mathbb{E}\left[Q_n(\tau)\right] < \infty.$$

Consider a rate vector $\boldsymbol{\lambda}$ interior to the inner capacity region Λ_{int} in Theorem 3.8. Namely, there exists an $\epsilon > 0$ and a probability distribution $\{\beta_{\boldsymbol{\phi}}\}_{\boldsymbol{\phi}\in\Phi}$ such that

$$\lambda_n + \epsilon < \sum_{\boldsymbol{\phi}\in\Phi} \beta_{\boldsymbol{\phi}}\, \eta_n^{\boldsymbol{\phi}}, \quad \forall\, n \in \{1, \dots, N\}, \tag{3.14}$$

where $\eta_n^{\boldsymbol{\phi}}$ is defined in (3.10). By Theorem 3.8, there exists a RandRR policy that yields the service rate over channel $n \in \{1, \dots, N\}$ equal to the right side of (3.14); thus, this policy stabilizes the network with arrival rate vector $\boldsymbol{\lambda}$ [Georgiadis et al. 2006, Lemma 3.6]. The existence of this policy is useful, and we denote it by RandRR^*.

Recall from Section 3.3 that the RandRR^* policy is defined by a probability distribution $\{\alpha_{\boldsymbol{\phi}}\}_{\boldsymbol{\phi}\in\Phi}$ over all $(2^N - 1)$ nonempty subset of channels. We may compute the RandRR^* policy to support a given data rate vector $\boldsymbol{\lambda}$. However, solving the optimal probabilities $\{\alpha_{\boldsymbol{\phi}}\}_{\boldsymbol{\phi}\in\Phi}$ associated with RandRR^* is difficult when N is large. It is because we need to find $(2^N - 1)$ unknown probabilities $\{\alpha_{\boldsymbol{\phi}}\}_{\boldsymbol{\phi}\in\Phi}$, compute $\{\beta_{\boldsymbol{\phi}}\}_{\boldsymbol{\phi}\in\Phi}$ from (3.21), and make (3.14) hold. This difficulty motivates us to construct the following simple queue-dependent policy.

Queue-Dependent Round Robin Policy QRR:

1. Start with $t = 0$.

2. At time t, observe the current queue backlog vector $\boldsymbol{Q}(t)$ and find the binary vector $\boldsymbol{\phi}(t) \in \Phi$ defined as [8]

$$\boldsymbol{\phi}(t) \triangleq \arg\max_{\boldsymbol{\phi}\in\Phi} f(\boldsymbol{Q}(t), \mathsf{RR}(\boldsymbol{\phi})), \tag{3.15}$$

[8] The vector $\boldsymbol{\phi}(t)$ is a queue-dependent decision. Thus we should write $\boldsymbol{\phi}(\boldsymbol{Q}(t), t)$ as a function of $\boldsymbol{Q}(t)$. For notational simplicity, we use $\boldsymbol{\phi}(t)$ instead.

where

$$f(\boldsymbol{Q}(t), \mathsf{RR}(\boldsymbol{\phi})) \triangleq \sum_{n:\phi_n=1} \left[Q_n(t)\mathbb{E}\left[L_{1n}^{\boldsymbol{\phi}} - 1\right] - \mathbb{E}\left[L_{1n}^{\boldsymbol{\phi}}\right] \sum_{n=1}^{N} Q_n(t)\lambda_n \right]$$

and $\mathbb{E}\left[L_{1n}^{\boldsymbol{\phi}}\right] = 1 + \mathsf{P}_{n,01}^{(M(\boldsymbol{\phi}))}/\mathsf{P}_{n,10}$ from (3.9). We emphasize that the summation in $f(\boldsymbol{Q}(t), \mathsf{RR}(\boldsymbol{\phi}))$ is only with respect to active channels in $\boldsymbol{\phi}$. Ties are broken arbitrarily.[9]

3. Run $\mathsf{RR}(\boldsymbol{\phi}(t))$ for one round of transmission. We emphasize that active channels in $\boldsymbol{\phi}(t)$ are served in the order of least recently used first. After the round ends, go to Step 2.

The QRR policy is a frame-based algorithm similar to RandRR, except that, at the beginning of every transmission round, the selection of a new subset of channels is no longer random but based on a queue-dependent rule. The QRR policy is of low complexity because we can compute $\boldsymbol{\phi}(t)$ in (3.15) in polynomial time with the following divide-and-conquer approach:

Solving Step 2 of QRR:

1. Partition the set Φ into subsets $\{\Phi_1, \ldots, \Phi_N\}$, where Φ_M, $M \in \{1, \ldots, N\}$, is the set of N-dimensional binary vectors having exactly M entries be 1.

2. For each $M \in \{1, \ldots, N\}$, find the maximizer of $f(\boldsymbol{Q}(t), \mathsf{RR}(\boldsymbol{\phi}))$ among vectors in Φ_M. Specifically, for each $\boldsymbol{\phi} \in \Phi_M$, we have

$$f(\boldsymbol{Q}(t), \mathsf{RR}(\boldsymbol{\phi})) = \sum_{n:\phi_n=1} \left[Q_n(t)\frac{\mathsf{P}_{n,01}^{(M)}}{\mathsf{P}_{n,10}} - \left(1 + \frac{\mathsf{P}_{n,01}^{(M)}}{\mathsf{P}_{n,10}}\right) \sum_{n=1}^{N} Q_n(t)\lambda_n \right]. \tag{3.16}$$

[9] In (3.52), we show that as long as the queue backlog vector $\boldsymbol{Q}(t)$ is not identically zero and the arrival rate vector $\boldsymbol{\lambda}$ is interior to the inner capacity bound Λ_{int}, we always have $\max_{\boldsymbol{\phi}\in\Phi} f(\boldsymbol{Q}(t), \mathsf{RR}(\boldsymbol{\phi})) > 0$.

Then, the maximizer of $f(\boldsymbol{Q}(t), \mathsf{RR}(\boldsymbol{\phi}))$ *activates* the M channels that yield the M largest summands at the right side of (3.16).

3. Obtain $\boldsymbol{\phi}(t)$ by comparing the maximizers from Step 2 for different values of $M \in \{1, \ldots, N\}$.

The detailed implementation is as follows.

Polynomial Time Implementation of Step 2 of QRR:

1. For each fixed $M \in \{1, \ldots, N\}$, we compute

$$Q_n(t) \frac{\mathsf{P}_{n,01}^{(M)}}{\mathsf{P}_{n,10}} - \left(1 + \frac{\mathsf{P}_{n,01}^{(M)}}{\mathsf{P}_{n,10}}\right) \sum_{n=1}^{N} Q_n(t) \lambda_n \tag{3.17}$$

   for all $n \in \{1, \ldots, N\}$. Sort these N numbers and define the binary vector $\boldsymbol{\phi}^M = (\phi_1^M, \ldots, \phi_N^M)$ such that $\phi_n^M = 1$ if the value (3.17) of channel n is among the M largest, otherwise $\phi_n^M = 0$. Ties are broken arbitrarily. Let $\hat{f}(\boldsymbol{Q}(t), M)$ denote the sum of the M largest values of (3.17).

2. Define $M(t) \triangleq \arg\max_{M \in \{1,\ldots,N\}} \hat{f}(\boldsymbol{Q}(t), M)$. Then, we assign $\boldsymbol{\phi}(t) = \boldsymbol{\phi}^{M(t)}$.

Using a novel variable-length frame-based Lyapunov drift analysis, we show in the next theorem that QRR stabilizes the network with any arrival rate vector $\boldsymbol{\lambda}$ strictly within the inner capacity bound Λ_{int}. The idea is to compare QRR with the (unknown) policy RandRR^* that stabilizes $\boldsymbol{\lambda}$. We show that, in every transmission round, QRR finds a round robin policy $\mathsf{RR}(\boldsymbol{\phi}(t))$ that yields a larger negative drift on queue backlogs than RandRR^* does. Therefore, QRR is stable.

Theorem 3.17 (Proof in Section 3.8.8)**.** For any data rate vector $\boldsymbol{\lambda}$ interior to Λ_{int}, policy QRR strongly stabilizes the network.

3.6 Chapter Summary and Discussions

To address how channel memory improves network throughput, in this chapter we study the network capacity region over Markov ON/OFF channels without the knowledge of current channel states. While solving the original problem is difficult because it is a restless multi-armed bandit, we construct an inner and an outer bound on the network capacity region, with the aid of channel memory. When channels are symmetric and the network serves a large number of users, we show the inner and outer bound are progressively tight when the data rates of different users are more balanced. We derive a simple queue-dependent round robin policy, which depends on data arrival rates and channel statistics, and show that this policy stabilizes the network for any data rate vector strictly within the inner capacity bound.

Transmitting data without channel probing is one of the many options for communication over a wireless network. Practically, each option may have pros and cons on criteria such as achievable throughput, power efficiency, implementation complexity, etc. It will be interesting explore how to combine all possible options to push the practically achievable network throughput to the limit. It will also be interesting to extend our method to more general cases, such as allowing limited channel probing or incorporating other performance measures such as energy consumption.

3.7 Bibliographic Notes

Scheduling over Markov ON/OFF channels with unknown current states is an analytical model for opportunistic spectrum access in cognitive radio networks (see Zhao and Swami [2007] and references therein), where the channel occupancy of a primary user acts as a Markov ON/OFF channel to the secondary users. It is also known as a restless multi-armed bandit problem [Whittle 1988]. In prior work, Ahmad and Liu [2009], Ahmad et al. [2009], Guha et al. [2009], Liu and Zhao [2010], Niño-Mora [2008], Zhao et al. [2008] study the maximization of sum of time average or discounted throughput (rewards). Ahmad and Liu [2009], Ahmad et al. [2009], Zhao et al. [2008] use dynamic programming and coupling methods, and show greedy round robin policies are optimal in some special cases; both positively and negatively correlated channels are studied. Index policies such Whittle's index [Whittle 1988] are constructed in Liu and Zhao [2010], Niño-Mora [2008], and are shown to have good performance by simulations. A $(2+\epsilon)$-approximate algorithm is derived in Guha et al. [2009] based on duality methods.

This chapter focuses on an easy-to-use characterization of the network capacity region over Markovian channels with unknown current states. Our results prepare us to study more complicated nonlinear utility maximization problems, for which the first step is to characterize the performance region. Utility maximization problems are addressed in the next chapter.

Jagannathan et al. [2011] characterizes the network capacity region over Markov ON/OFF channels with unknown current states as a limit of a sequence of linear programs, each of which solves a finite-state Markov decision process (MDP) truncated from the ideal MDP, which has a countably infinite state space, that describes the full network capacity region. A throughput-optimal frame-based policy is provided. Due to the curse of dimensionality, this approach may not scale well with the number of channels.

This chapter is also a study on efficient scheduling over wireless networks with delayed/uncertain channel state information (CSI); see Pantelidou et al. [2007], Ying and Shakkottai [2008, 2009] and references therein. The work on delayed CSI that is most closely related to ours is Ying and Shakkottai [2008, 2009], where the authors study the capacity region and throughput-optimal policies of different wireless networks, assuming that channel states are *persistently* probed but fed back *with delay*. We note that our results are significantly different. Here channels are never probed, and new (delayed) channel state information is only acquired when the channel is served. In other words, acquiring the delayed channel state information is part of the control decisions in this chapter. Our results also apply to an important scenario in partial channel probing (see Chaporkar et al. [2009], Li and Neely [2010a] and references therein) where at most one channel is probed in every slot, and data can only be served over the probed channel but not on unknown ones. As far as throughput is concerned and that we neglect probing overhead, this scenario is equivalent to blindly transmitting data over a channel in every slot. Different from previous work which usually assumes channels are i.i.d. over slots, here we show how channel memory improves throughput under a limited probing regime.

3.8 Proofs in Chapter 3

3.8.1 Proof of Lemma 3.2

Initially, by (3.3) we have $\omega_n(0) = \pi_{n,\mathsf{ON}} \geq \mathsf{P}_{n,01}^{(M)}$ for all n. Suppose the base station switches to channel n at time t, and the last previous use of channel n was at slot $(t-k)$ for some $k < t$. In slot $(t-k)$, there are two possible cases:

1. Channel n turns OFF, and as a result the information state on slot t is $\omega_n(t) = \mathsf{P}_{n,01}^{(k)}$. Due to round robin, the other $(M-1)$ channels must have been used for at least one slot before t after slot $(t-k)$, and thus $k \geq M$. By (3.3) we have $\omega_n(t) = \mathsf{P}_{n,01}^{(k)} \geq \mathsf{P}_{n,01}^{(M)}$.

2. Channel n is ON and transmits a dummy packet. Thus $\omega_n(t) = \mathsf{P}_{n,11}^{(k)}$. By (3.3) we have $\omega_n(t) = \mathsf{P}_{n,11}^{(k)} \geq \mathsf{P}_{n,01}^{(M)}$. $\square$

3.8.2 Proof of Lemma 3.6

At the beginning of a new round, suppose round robin policy $\mathsf{RR}(\boldsymbol{\phi})$ is selected. We index the $M(\boldsymbol{\phi})$ active channels in $\boldsymbol{\phi}$ as $(n_1, n_2, \dots, n_{M(\boldsymbol{\phi})})$, which is in the decreasing order of the time duration between their last use and the beginning of the current round. In other words, the last use of n_k is earlier than that of $n_{k'}$ only if $k < k'$. Fix an active channel n_k. Then it suffices to show that when this channel is served in the current round, the time duration back to the end of its last service is at least $(M(\boldsymbol{\phi}) - 1)$ slots (that this channel has information state no worse than $\mathsf{P}_{n_k,01}^{(M(\boldsymbol{\phi}))}$ then follows the same arguments in the proof of Lemma 3.2).

We partition the active channels in $\boldsymbol{\phi}$ other than n_k into two sets $\mathcal{A} = \{n_1, n_2, \dots, n_{k-1}\}$ and $\mathcal{B} = \{n_{k+1}, n_{k+2}, \dots, n_{M(\boldsymbol{\phi})}\}$. Then the last use of every channel in $\mathcal{B}$ occurs after the last use of n_k, and so channel n_k has been idled for at least $|\mathcal{B}|$ slots at the start of the current round. However, the policy in this round will serve all channels in $\mathcal{A}$ before serving n_k, taking at least one slot per channel, and so we wait at least additional $|\mathcal{A}|$ slots before serving channel n_k. The total time that this channel has been idled is thus at least $|\mathcal{A}| + |\mathcal{B}| = M(\boldsymbol{\phi}) - 1$. $\square$

3.8.3 Proof of Theorem 3.8

Let $Z(t)$ denote the number of times Step 1 of RandRR is executed in $[0, t)$, in which we suppose vector $\boldsymbol{\phi}$ is selected $Z_{\boldsymbol{\phi}}(t)$ times. Define t_i, where $i \in \mathbb{Z}^+$, as the $(i+1)$th time instant a new vector $\boldsymbol{\phi}$ is selected. Assume $t_0 = 0$, and thus the first selection occurs at time 0. It follows that $Z(t_i^-) = i$, $Z(t_i) = i+1$, and the ith round of packet transmissions ends at time t_i^-.

Fix a vector $\boldsymbol{\phi}$. Within the time periods in which policy $\mathsf{RR}(\boldsymbol{\phi})$ is executed, denote by $L_{kn}^{\boldsymbol{\phi}}$ the duration of the kth time the base station stays with channel n. The time average throughput that policy $\mathsf{RR}(\boldsymbol{\phi})$ yields on its active channel n over $[0, t_i)$ is

$$\frac{\sum_{k=1}^{Z_{\boldsymbol{\phi}}(t_i)} \left(L_{kn}^{\boldsymbol{\phi}} - 1\right)}{\sum_{\boldsymbol{\phi}\in\Phi} \sum_{k=1}^{Z_{\boldsymbol{\phi}}(t_i)} \sum_{n:\phi_n=1} L_{kn}^{\boldsymbol{\phi}}}. \tag{3.18}$$

For simplicity, we focus on discrete time instants $\{t_i\}$ large enough so that $Z_{\boldsymbol{\phi}}(t_i) > 0$ for all $\boldsymbol{\phi} \in \Phi$ (so that the sums in (3.18) make sense). The generalization to arbitrary time t can be done by incorporating fractional transmission rounds, which are amortized over time. Next, rewrite (3.18) as

$$\frac{\sum_{k=1}^{Z_{\boldsymbol{\phi}}(t_i)} \sum_{n:\phi_n=1} L_{kn}^{\boldsymbol{\phi}}}{\sum_{\boldsymbol{\phi}\in\Phi} \sum_{k=1}^{Z_{\boldsymbol{\phi}}(t_i)} \sum_{n:\phi_n=1} L_{kn}^{\boldsymbol{\phi}}} \underbrace{\frac{\sum_{k=1}^{Z_{\boldsymbol{\phi}}(t_i)} \left(L_{kn}^{\boldsymbol{\phi}} - 1\right)}{\sum_{k=1}^{Z_{\boldsymbol{\phi}}(t_i)} \sum_{n:\phi_n=1} L_{kn}^{\boldsymbol{\phi}}}}_{(*)}. \tag{3.19}$$

As $t \to \infty$, the second term $(*)$ of (3.19) satisfies

$$(*) = \frac{\frac{1}{Z_{\boldsymbol{\phi}}(t_i)} \sum_{k=1}^{Z_{\boldsymbol{\phi}}(t_i)} \left(L_{kn}^{\boldsymbol{\phi}} - 1\right)}{\sum_{n:\phi_n=1} \frac{1}{Z_{\boldsymbol{\phi}}(t_i)} \sum_{k=1}^{Z_{\boldsymbol{\phi}}(t_i)} L_{kn}^{\boldsymbol{\phi}}} \overset{(a)}{\to} \frac{\mathbb{E}\left[L_{1n}^{\boldsymbol{\phi}} - 1\right]}{\sum_{n:\phi_n=1} \mathbb{E}\left[L_{1n}^{\boldsymbol{\phi}}\right]} \overset{(b)}{=} \eta_n^{\boldsymbol{\phi}},$$

where (a) is by the Law of Large Numbers (we have shown in Corollary 3.7 that $L_{kn}^{\boldsymbol{\phi}}$ are i.i.d. for different k) and (b) by (3.10).

Denote the first term of (3.19) by $\beta_{\boldsymbol{\phi}}(t_i)$. We note that $\beta_{\boldsymbol{\phi}}(t_i) \in [0, 1]$ for all $\boldsymbol{\phi} \in \Phi$ and $\sum_{\boldsymbol{\phi}\in\Phi} \beta_{\boldsymbol{\phi}}(t_i) = 1$. We rewrite $\beta_{\boldsymbol{\phi}}(t_i)$ as

$$\beta_{\boldsymbol{\phi}}(t_i) = \frac{\left[\frac{Z_{\boldsymbol{\phi}}(t_i)}{Z(t_i)}\right] \sum_{n:\phi_n=1} \left[\frac{1}{Z_{\boldsymbol{\phi}}(t_i)} \sum_{k=1}^{Z_{\boldsymbol{\phi}}(t_i)} L_{kn}^{\boldsymbol{\phi}}\right]}{\sum_{\boldsymbol{\phi}\in\Phi} \left[\frac{Z_{\boldsymbol{\phi}}(t_i)}{Z(t_i)}\right] \sum_{n:\phi_n=1} \left[\frac{1}{Z_{\boldsymbol{\phi}}(t_i)} \sum_{k=1}^{Z_{\boldsymbol{\phi}}(t_i)} L_{kn}^{\boldsymbol{\phi}}\right]}.$$

As $t \to \infty$, we have

$$\beta_{\boldsymbol{\phi}} \triangleq \lim_{i\to\infty} \beta_{\boldsymbol{\phi}}(t_i) = \frac{\alpha_{\boldsymbol{\phi}} \sum_{n:\phi_n=1} \mathbb{E}\left[L_{1n}^{\boldsymbol{\phi}}\right]}{\sum_{\boldsymbol{\phi}\in\Phi} \alpha_{\boldsymbol{\phi}} \sum_{n:\phi_n=1} \mathbb{E}\left[L_{1n}^{\boldsymbol{\phi}}\right]}, \tag{3.20}$$

which uses, by the Law of Large Numbers,

$$\frac{Z_{\boldsymbol{\phi}}(t_i)}{Z(t_i)} \to \alpha_{\boldsymbol{\phi}}, \quad \frac{1}{Z_{\boldsymbol{\phi}}(t_i)} \sum_{k=1}^{Z_{\boldsymbol{\phi}}(t_i)} L_{kn}^{\boldsymbol{\phi}} \to \mathbb{E}\left[L_{1n}^{\boldsymbol{\phi}}\right].$$

From (3.18)(3.19)(3.20), we have shown that the throughput contributed by policy $\mathsf{RR}(\boldsymbol{\phi})$ on its active channel n is $\beta_{\boldsymbol{\phi}}\, \eta_n^{\boldsymbol{\phi}}$. Consequently, RandRR parameterized by $\{\alpha_{\boldsymbol{\phi}}\}_{\boldsymbol{\phi}\in\Phi}$

supports any data rate vector $\boldsymbol{\lambda}$ that is entrywise dominated by $\boldsymbol{\lambda} \leq \sum_{\boldsymbol{\phi}\in\Phi} \beta_{\boldsymbol{\phi}}\, \boldsymbol{\eta}^{\boldsymbol{\phi}}$, where $\{\beta_{\boldsymbol{\phi}}\}_{\boldsymbol{\phi}\in\Phi}$ is defined in (3.20) and $\boldsymbol{\eta}^{\boldsymbol{\phi}}$ in (3.10).

The above analysis shows that every RandRR policy achieves a boundary point of Λ_{int} defined in Theorem 3.8. Conversely, the next lemma, proved in Section 3.8.4, shows that every boundary point of Λ_{int} is achievable by some RandRR policy, and the proof is complete.

Lemma 3.18. For any probability distribution $\{\beta_{\boldsymbol{\phi}}\}_{\boldsymbol{\phi}\in\Phi}$, there exists another probability distribution $\{\alpha_{\boldsymbol{\phi}}\}_{\boldsymbol{\phi}\in\Phi}$ that solves the linear system

$$\beta_{\boldsymbol{\phi}} = \frac{\alpha_{\boldsymbol{\phi}} \sum_{n:\phi_n=1} \mathbb{E}\left[L_{1n}^{\boldsymbol{\phi}}\right]}{\sum_{\boldsymbol{\phi}\in\Phi} \alpha_{\boldsymbol{\phi}} \sum_{n:\phi_n=1} \mathbb{E}\left[L_{1n}^{\boldsymbol{\phi}}\right]}, \quad \text{for all } \boldsymbol{\phi} \in \Phi. \tag{3.21}$$

□

3.8.4 Proof of Lemma 3.18

For any probability distribution $\{\beta_{\boldsymbol{\phi}}\}_{\boldsymbol{\phi}\in\Phi}$, we prove the lemma by inductively constructing the solution $\{\alpha_{\boldsymbol{\phi}}\}_{\boldsymbol{\phi}\in\Phi}$ to (3.21). The induction is on the cardinality of Φ. Without loss of generality, we index elements in Φ by $\Phi = \{\boldsymbol{\phi}^1, \boldsymbol{\phi}^2, \dots\}$, where $\boldsymbol{\phi}^k = (\phi_1^k, \dots, \phi_N^k)$. We define $\chi_k \triangleq \sum_{n:\phi_n^k=1} \mathbb{E}\left[L_{1n}^{\boldsymbol{\phi}^k}\right]$, $\beta_k \triangleq \beta_{\boldsymbol{\phi}^k}$, and $\alpha_k \triangleq \alpha_{\boldsymbol{\phi}^k}$. Then, we rewrite (3.21) as

$$\beta_k = \frac{\alpha_k \chi_k}{\sum_{1\leq k\leq|\Phi|} \alpha_k \chi_k}, \quad \text{for all } k \in \{1, 2, \dots, |\Phi|\}. \tag{3.22}$$

We first note that $\Phi = \{\boldsymbol{\phi}^1\}$ is a degenerate case where β_1 and α_1 must both be 1. When $\Phi = \{\boldsymbol{\phi}^1, \boldsymbol{\phi}^2\}$, for any probability distribution $\{\beta_1, \beta_2\}$ with *positive elements*,[10] it is easy to show

$$\alpha_1 = \frac{\chi_2 \beta_1}{\chi_1 \beta_2 + \chi_2 \beta_1}, \quad \alpha_2 = 1 - \alpha_1.$$

[10] If one element of $\{\beta_1, \beta_2\}$ is zero, say $\beta_2 = 0$, we can show necessarily $\alpha_2 = 0$ and it degenerates to the one-policy case $\Phi = \{\boldsymbol{\phi}^1\}$. Such degeneration also happens generally. Thus, without loss of generality, in the rest of the proof we only consider probability distributions that only have positive elements.

Let $\Phi = \{\boldsymbol{\phi}^k : 1 \leq k \leq K\}$ for some $K \geq 2$. Assume that, for any probability distribution $\{\beta_k > 0 : 1 \leq k \leq K\}$, we can find $\{\alpha_k : 1 \leq k \leq K\}$ that solves (3.22).

For the case $\Phi = \{\boldsymbol{\phi}^k : 1 \leq k \leq K+1\}$ and any $\{\beta_k > 0 : 1 \leq k \leq K+1\}$, we construct the solution $\{\alpha_k : 1 \leq k \leq K+1\}$ to (3.22) as follows. Let $\{\gamma_2, \gamma_3, \ldots, \gamma_{K+1}\}$ be the solution to the linear system

$$\frac{\gamma_k \chi_k}{\sum_{k=2}^{K+1} \gamma_k \chi_k} = \frac{\beta_k}{\sum_{k=2}^{K+1} \beta_k}, \quad 2 \leq k \leq K+1. \tag{3.23}$$

By the induction assumption, the set $\{\gamma_2, \gamma_3, \ldots, \gamma_{K+1}\}$ exists and satisfies $\gamma_k \in [0,1]$ for $2 \leq k \leq K+1$ and $\sum_{k=2}^{K+1} \gamma_k = 1$. Define

$$\alpha_1 \triangleq \frac{\beta_1 \sum_{k=2}^{K+1} \gamma_k \chi_k}{\chi_1(1-\beta_1) + \beta_1 \sum_{k=2}^{K+1} \gamma_k \chi_k} \tag{3.24}$$

$$\alpha_k \triangleq (1-\alpha_1)\gamma_k, \quad 2 \leq k \leq K+1. \tag{3.25}$$

It remains to show (3.24) and (3.25) are the desired solution. It is easy to observe that $\alpha_k \in [0,1]$ for $1 \leq k \leq K+1$, and

$$\sum_{k=1}^{K+1} \alpha_k = \alpha_1 + (1-\alpha_1) \sum_{k=2}^{K+1} \gamma_k = \alpha_1 + (1-\alpha_1) = 1.$$

By rearranging terms in (3.24) and using (3.25), we have

$$\beta_1 = \frac{\alpha_1 \chi_1}{\alpha_1 \chi_1 + \sum_{k=2}^{K+1} (1-\alpha_1)\gamma_k \chi_k} = \frac{\alpha_1 \chi_1}{\sum_{k=1}^{K+1} \alpha_k \chi_k}. \tag{3.26}$$

In addition, for $2 \le k \le K+1$, we have

$$\begin{aligned}
\frac{\alpha_k \chi_k}{\sum_{k=1}^{K+1} \alpha_k \chi_k} &= \left[\frac{\alpha_k \chi_k}{\sum_{k=2}^{K+1} \alpha_k \chi_k}\right] \left[\frac{\sum_{k=2}^{K+1} \alpha_k \chi_k}{\sum_{k=1}^{K+1} \alpha_k \chi_k}\right] \\
&\overset{(a)}{=} \left[\frac{(1-\alpha_1)\gamma_k \chi_k}{\sum_{k=2}^{K+1} (1-\alpha_1)\gamma_k \chi_k}\right] \left[1 - \frac{\alpha_1 \chi_1}{\sum_{k=1}^{K+1} \alpha_k \chi_k}\right] \\
&\overset{(b)}{=} \left[\frac{\gamma_k \chi_k}{\sum_{k=2}^{K+1} \gamma_k \chi_k}\right] (1-\beta_1) \\
&\overset{(c)}{=} \left(\frac{\beta_k}{\sum_{k=2}^{K+1} \beta_k}\right) (1-\beta_1) \\
&\overset{(d)}{=} \beta_k,
\end{aligned}$$

where (a) is by plugging in (3.25), (b) uses (3.26), (c) uses (3.23), and (d) is by $\sum_{k=1}^{K+1} \beta_k = 1$. The proof is complete. □

3.8.5 Proof of Lemma 3.10

Let $\mathcal{N}_1(T) \subseteq \{0, 1, \ldots, T-1\}$ be the subset of time instants in which $Y(t) = 1$. Note that $\sum_{t=0}^{T-1} Y(t) = |\mathcal{N}_1(T)|$. For each $t \in \mathcal{N}_1(T)$, let $1_{[1\to 0]}(t)$ be an indicator function which is 1 if $Y(t)$ transits from 1 to 0 at time t, and 0 otherwise. We define $\mathcal{N}_0(T)$ and $1_{[0\to 1]}(t)$ similarly.

In $\{0, 1, \ldots, T-1\}$, since state transitions of $\{Y(t)\}$ from 1 to 0 and from 0 to 1 differ by at most 1, we have

$$\left| \sum_{t\in\mathcal{N}_1(T)} 1_{[1\to 0]}(t) - \sum_{t\in\mathcal{N}_0(T)} 1_{[0\to 1]}(t) \right| \le 1, \tag{3.27}$$

which is true for all T. Dividing (3.27) by T, we get

$$\left| \frac{1}{T} \sum_{t\in\mathcal{N}_1(T)} 1_{[1\to 0]}(t) - \frac{1}{T} \sum_{t\in\mathcal{N}_0(T)} 1_{[0\to 1]}(t) \right| \le \frac{1}{T}. \tag{3.28}$$

Consider the subsequence $\{T_k\}$ such that

$$\lim_{k\to\infty} \frac{1}{T_k} \sum_{t=0}^{T_k-1} Y(t) = \pi_Y(1) = \lim_{k\to\infty} \frac{|\mathcal{N}_1(T_k)|}{T_k}. \tag{3.29}$$

Note that $\{T_k\}$ exists because $\{(1/T)\sum_{t=0}^{T-1} Y(t)\}_T$ is a bounded sequence indexed by integers T. Moreover, there exists a subsequence $\{T_n\}$ of $\{T_k\}$ so that each of the two averages in (3.28) has a limit point with respect to $\{T_n\}$, because they are bounded sequences, too. In the rest of the proof, we will work on $\{T_n\}$, but we drop subscript n for notational simplicity. Passing $T\to\infty$, we get from (3.28) that

$$\begin{aligned}
&\underbrace{\left(\lim_{T\to\infty} \frac{|\mathcal{N}_1(T)|}{T}\right)}_{\overset{(a)}{=}\pi_Y(1)} \underbrace{\left(\lim_{T\to\infty} \frac{1}{|\mathcal{N}_1(T)|} \sum_{t\in\mathcal{N}_1(T)} 1_{[1\to 0]}(t)\right)}_{\triangleq \beta} \\
&= \underbrace{\left(\lim_{T\to\infty} \frac{|\mathcal{N}_0(T)|}{T}\right)}_{\overset{(b)}{=}1-\pi_Y(1)} \underbrace{\left(\lim_{T\to\infty} \frac{1}{|\mathcal{N}_0(T)|} \sum_{t\in\mathcal{N}_0(T)} 1_{[0\to 1]}(t)\right)}_{\triangleq \gamma},
\end{aligned} \tag{3.30}$$

where (a) is by (3.29) and (b) is by $|\mathcal{N}_1(T)| + |\mathcal{N}_0(T)| = T$. From (3.30) we get

$$\pi_Y(1) = \frac{\gamma}{\beta+\gamma}.$$

The next lemma, proved in Section 3.8.6, helps to show $\gamma \le \mathsf{P}_{01}$. □

Lemma 3.19 (Stochastic Coupling of Random Binary Sequences)**.** Let $\{I_n\}_{n=1}^{\infty}$ be an infinite sequence of binary random variables. Suppose for all $n \in \{1, 2, \ldots\}$ we have

$$\Pr\left[I_n = 1 \mid I_1 = i_1, \ldots, I_{n-1} = i_{n-1}\right] \le \mathsf{P}_{01} \tag{3.31}$$

for all possible values of $i_1, \ldots, i_{n-1}$. Then, we can construct a new sequence $\{\hat{I}_n\}_{n=1}^{\infty}$ of binary random variables that are i.i.d. with $\Pr\left[\hat{I}_n = 1\right] = \mathsf{P}_{01}$ for all n and satisfy $\hat{I}_n \geq I_n$ for all n. Consequently, we have

$$\limsup_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} I_n \leq \limsup_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \hat{I}_n = \mathsf{P}_{01}.$$

To use Lemma 3.19 to prove $\gamma \leq \mathsf{P}_{01}$, we let t_n denote the nth time $Y(t) = 0$ and let $I_n = 1_{[0\to 1]}(t_n)$. For simplicity, we assume $\{t_n\}$ is an infinite sequence so that state 0 is visited infinitely often in $\{Y(t)\}$. By the assumption that $\mathsf{Q}_{01}(t) \leq \mathsf{P}_{01}$ for all t, we know (3.31) holds. Therefore, Lemma 3.19 yields that

$$\gamma \leq \limsup_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} 1_{[0\to 1]}(t_n) \leq \mathsf{P}_{01}.$$

Similarly as Lemma 3.19, we can show $\beta \geq \mathsf{P}_{10}$ by stochastic coupling. Therefore

$$\pi_Y(1) = \frac{\gamma}{\beta + \gamma} \leq \frac{\gamma}{\mathsf{P}_{10} + \gamma} \leq \frac{\mathsf{P}_{01}}{\mathsf{P}_{01} + \mathsf{P}_{10}} = \pi_X(1). \quad \Box$$

3.8.6 Proof of Lemma 3.19

For simplicity, we assume

$$\Pr\left[I_n = 0 \mid I_1 = i_1, \ldots, I_{n-1} = i_{n-1}\right] > 0$$

for all n and all possible values of $i_1, \ldots, i_{n-1}$. For each $n \in \{1, 2, \ldots\}$, define $\hat{I}_n$ as follows: If $I_n = 1$, define $\hat{I}_n = 1$. If $I_n = 0$, observe the *history* $I_1^{n-1} \triangleq (I_1, \ldots, I_{n-1})$ and independently choose $\hat{I}_n$ as follows:

$$\hat{I}_n = \begin{cases} 1 & \text{with prob. } \frac{\mathsf{P}_{01} - \Pr\left[I_n = 1 | I_1^{n-1}\right]}{\Pr\left[I_n = 0 | I_1^{n-1}\right]} \\ 0 & \text{with prob. } 1 - \frac{\mathsf{P}_{01} - \Pr\left[I_n = 1 | I_1^{n-1}\right]}{\Pr\left[I_n = 0 | I_1^{n-1}\right]}. \end{cases} \tag{3.32}$$

The probabilities in (3.32) are well-defined because $\mathsf{P}_{01} \geq \Pr\left[I_n = 1 \mid I_1^{n-1}\right]$ by (3.31), and

$$\mathsf{P}_{01} \leq 1 = \Pr\left[I_n = 1 \mid I_1^{n-1}\right] + \Pr\left[I_n = 0 \mid I_1^{n-1}\right].$$

Therefore

$$\mathsf{P}_{01} - \Pr\left[I_n = 1 \mid I_1^{n-1}\right] \leq \Pr\left[I_n = 0 \mid I_1^{n-1}\right].$$

With the above definition of $\hat{I}_n$, we have $\hat{I}_n = 1$ whenever $I_n = 1$. Therefore $\hat{I}_n \geq I_n$ for all n. Furthermore, for any n and any binary vector $i_1^{n-1} \triangleq (i_1, \ldots, i_{n-1})$, we have, by definition of $\hat{I}_n$,

$$\begin{aligned}
&\Pr\left[\hat{I}_n = 1 \mid I_1^{n-1} = i_1^{n-1}\right] \\
&= \Pr\left[I_n = 1 \mid I_1^{n-1} = i_1^{n-1}\right] + \Pr\left[I_n = 0 \mid I_1^{n-1} = i_1^{n-1}\right] \\
&\quad \times \frac{\mathsf{P}_{01} - \Pr\left[I_n = 1 \mid I_1^{n-1} = i_1^{n-1}\right]}{\Pr\left[I_n = 0 \mid I_1^{n-1} = i_1^{n-1}\right]} \\
&= \mathsf{P}_{01}.
\end{aligned} \tag{3.33}$$

Therefore, for all n we have

$$\Pr\left[\hat{I}_n = 1\right] = \sum_{i_1^{n-1}} \Pr\left[\hat{I}_n = 1 \mid I_1^{n-1} = i_1^{n-1}\right] \Pr\left[I_1^{n-1} = i_1^{n-1}\right] = \mathsf{P}_{01}.$$

Thus, the $\hat{I}_n$ variables are identically distributed. It remains to prove that they are independent.

Suppose components in $\hat{I}_1^n \triangleq (\hat{I}_1, \ldots, \hat{I}_n)$ are independent. We prove that components in $\hat{I}_1^{n+1} = (\hat{I}_1, \ldots, \hat{I}_{n+1})$ are also independent. For any binary vector $\hat{i}_1^{n+1} \triangleq (\hat{i}_1, \ldots, \hat{i}_{n+1})$, since

$$\begin{aligned}
\Pr\left[\hat{I}_1^{n+1} = \hat{i}_1^{n+1}\right] &= \Pr\left[\hat{I}_{n+1} = \hat{i}_{n+1} \mid \hat{I}_1^n = \hat{i}_1^n\right] \Pr\left[\hat{I}_1^n = \hat{i}_1^n\right] \\
&= \Pr\left[\hat{I}_{n+1} = \hat{i}_{n+1} \mid \hat{I}_1^n = \hat{i}_1^n\right] \prod_{k=1}^{n} \Pr\left[\hat{I}_k = \hat{i}_k\right],
\end{aligned}$$

it suffices to show

$$\Pr\left[\hat{I}_{n+1} = 1 \mid \hat{I}_1^n = \hat{i}_1^n\right] = \Pr\left[\hat{I}_{n+1} = 1\right] = \mathsf{P}_{01}.$$

Indeed,

$$\begin{aligned}
\Pr\left[\hat{I}_{n+1} = 1 \mid \hat{I}_1^n = \hat{i}_1^n\right] &= \sum_{i_1^n} \Pr\left[\hat{I}_{n+1} = 1 \mid I_1^n = i_1^n, \hat{I}_1^n = \hat{i}_1^n\right] \Pr\left[I_1^n = i_1^n \mid \hat{I}_1^n = \hat{i}_1^n\right] \\
&= \sum_{i_1^n} \Pr\left[\hat{I}_{n+1} = 1 \mid I_1^n = i_1^n\right] \Pr\left[I_1^n = i_1^n \mid \hat{I}_1^n = \hat{i}_1^n\right] \\
&\overset{(a)}{=} \sum_{i_1^n} \mathsf{P}_{01} \Pr\left[I_1^n = i_1^n \mid \hat{I}_1^n = \hat{i}_1^n\right] \\
&= \mathsf{P}_{01},
\end{aligned}$$

where (a) is by (3.33). The proof is complete. $\square$

3.8.7 Proof of Lemma 3.15

By definition of $d(\boldsymbol{v})$, there exists a nonempty subset $A \subseteq \Phi_{d(\boldsymbol{v})}$, and for every $\boldsymbol{\phi} \in A$ a positive real number $\hat{\beta}_{\boldsymbol{\phi}} > 0$, such that $\boldsymbol{v} = \sum_{\boldsymbol{\phi} \in A} \hat{\beta}_{\boldsymbol{\phi}} \boldsymbol{\phi}$. For each $\boldsymbol{\phi} \in A$, we have $M(\boldsymbol{\phi}) = d(\boldsymbol{v})$ and $c_{M(\boldsymbol{\phi})} = c_{d(\boldsymbol{v})}$. Define

$$\beta_{\boldsymbol{\phi}} \triangleq \frac{\hat{\beta}_{\boldsymbol{\phi}}}{\sum_{\boldsymbol{\phi} \in A} \hat{\beta}_{\boldsymbol{\phi}}}$$

for each $\boldsymbol{\phi} \in A$; $\{\beta_{\boldsymbol{\phi}}\}_{\boldsymbol{\phi}\in A}$ is a probability distribution. Consider a RandRR policy that, in every round, selects $\boldsymbol{\phi} \in A$ with probability $\beta_{\boldsymbol{\phi}}$. By Lemma 3.4, this RandRR policy achieves throughput vector $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_N)$ that satisfies

$$\begin{aligned}
\boldsymbol{\lambda} = \sum_{\boldsymbol{\phi}\in A} \beta_{\boldsymbol{\phi}} \frac{c_{M(\boldsymbol{\phi})}}{M(\boldsymbol{\phi})} \boldsymbol{\phi} = \frac{c_{d(\boldsymbol{v})}}{d(\boldsymbol{v})} \sum_{\boldsymbol{\phi}\in A} \frac{\hat{\beta}_{\boldsymbol{\phi}}}{\sum_{\boldsymbol{\phi}\in A} \hat{\beta}_{\boldsymbol{\phi}}} \boldsymbol{\phi} &= \frac{c_{d(\boldsymbol{v})}}{d(\boldsymbol{v}) \sum_{\boldsymbol{\phi}\in A} \hat{\beta}_{\boldsymbol{\phi}}} \sum_{\boldsymbol{\phi}\in A} \hat{\beta}_{\boldsymbol{\phi}} \boldsymbol{\phi} \\
&= \left(\frac{c_{d(\boldsymbol{v})}}{d(\boldsymbol{v}) \sum_{\boldsymbol{\phi}\in A} \hat{\beta}_{\boldsymbol{\phi}}} \right) \boldsymbol{v},
\end{aligned}$$

which is in the direction of $\boldsymbol{v}$. In addition, the sum throughput is equal to

$$\sum_{n=1}^{N} \lambda_n = \sum_{\boldsymbol{\phi}\in A} \beta_{\boldsymbol{\phi}} \frac{c_{M(\boldsymbol{\phi})}}{M(\boldsymbol{\phi})} \left(\sum_{n=1}^{N} \phi_n \right) = \sum_{\boldsymbol{\phi}\in A} \beta_{\boldsymbol{\phi}}\, c_{M(\boldsymbol{\phi})} = c_{d(\boldsymbol{v})}.$$

□

3.8.8 Proof of Theorem 3.17

(**A Related** RandRR **Policy**) For each randomized round robin policy RandRR, it is useful to consider a *renewal reward process* where renewal epochs are defined as time instants at which RandRR starts a new round of transmission.[11] Let T denote the renewal period. We say one unit of reward is earned by a user if RandRR serves a packet to that user. Let R_n denote the sum reward earned by user n in one renewal period T, representing the number of successful transmissions user n receives in one round of

[11] We note that the renewal reward process is defined solely with respect to RandRR, and is only used to facilitate our analysis. At these renewal epochs, the state of the network, including the current queue state $\boldsymbol{Q}(t)$, does *not* necessarily renew itself.

scheduling. Conditioning on the round robin policy $\mathsf{RR}(\boldsymbol{\phi})$ chosen by RandRR for the current round of transmission, we have from Corollary 3.7:

$$\mathbb{E}[T] = \sum_{\boldsymbol{\phi}\in\Phi} \alpha_{\boldsymbol{\phi}}\, \mathbb{E}[T \mid \mathsf{RR}(\boldsymbol{\phi})] \tag{3.34}$$

$$\mathbb{E}[T \mid \mathsf{RR}(\boldsymbol{\phi})] = \sum_{n:\phi_n=1} \mathbb{E}\left[L_{1n}^{\boldsymbol{\phi}}\right], \tag{3.35}$$

and, for all $n \in \{1, 2, \ldots, N\}$,

$$\mathbb{E}[R_n] = \sum_{\boldsymbol{\phi}\in\Phi} \alpha_{\boldsymbol{\phi}}\, \mathbb{E}[R_n \mid \mathsf{RR}(\boldsymbol{\phi})] \tag{3.36}$$

$$\mathbb{E}[R_n \mid \mathsf{RR}(\boldsymbol{\phi})] = \begin{cases} \mathbb{E}\left[L_{1n}^{\boldsymbol{\phi}} - 1\right] & \text{if } \phi_n = 1 \\ 0 & \text{if } \phi_n = 0. \end{cases} \tag{3.37}$$

Consider the round robin policy $\mathsf{RR}((1, 1, \ldots, 1))$ that serves all N channels in one round. We define $T_{\max}$ as its renewal period. From Corollary 3.7, we know $\mathbb{E}[T_{\max}] < \infty$ and $\mathbb{E}\left[(T_{\max})^2\right] < \infty$. Further, for any RandRR, including the one that use a fixed $\mathsf{RR}(\boldsymbol{\phi})$ policy in every round as a special case, we can show that $T_{\max}$ is stochastically larger than the renewal period T, and $(T_{\max})^2$ is stochastically larger than T^2. It follows that

$$\mathbb{E}[T] \le \mathbb{E}[T_{\max}], \ \mathbb{E}\left[T^2\right] \le \mathbb{E}\left[(T_{\max})^2\right]. \tag{3.38}$$

We have denoted by RandRR^* (in the discussion after (3.14)) the randomized round robin policy that yields a service rate vector strictly larger than the target arrival rate vector $\boldsymbol{\lambda}$ entrywise. Let T^* denote the renewal period of RandRR^*, and R_n^* the sum reward (the number of successful transmissions) received by user n over the renewal period T^*. Then, we have

$$\begin{aligned}
\frac{\mathbb{E}\left[R_n^*\right]}{\mathbb{E}\left[T^*\right]} &\overset{(a)}{=} \frac{\sum_{\boldsymbol{\phi}\in\Phi} \alpha_{\boldsymbol{\phi}}\, \mathbb{E}\left[R_n^* \mid \mathsf{RR}(\boldsymbol{\phi})\right]}{\sum_{\boldsymbol{\phi}\in\Phi} \alpha_{\boldsymbol{\phi}}\, \mathbb{E}\left[T^* \mid \mathsf{RR}(\boldsymbol{\phi})\right]} \\
&\overset{(b)}{=} \sum_{\boldsymbol{\phi}\in\Phi} \left(\frac{\alpha_{\boldsymbol{\phi}}}{\sum_{\boldsymbol{\phi}\in\Phi} \alpha_{\boldsymbol{\phi}}\, \mathbb{E}\left[T^* \mid \mathsf{RR}(\boldsymbol{\phi})\right]} \right) \mathbb{E}\left[R_n^* \mid \mathsf{RR}(\boldsymbol{\phi})\right] \\
&= \sum_{\boldsymbol{\phi}\in\Phi} \underbrace{\frac{\alpha_{\boldsymbol{\phi}}\, \mathbb{E}\left[T^* \mid \mathsf{RR}(\boldsymbol{\phi})\right]}{\sum_{\boldsymbol{\phi}\in\Phi} \alpha_{\boldsymbol{\phi}}\, \mathbb{E}\left[T^* \mid \mathsf{RR}(\boldsymbol{\phi})\right]}}_{(c)=\beta_n} \underbrace{\frac{\mathbb{E}\left[R_n^* \mid \mathsf{RR}(\boldsymbol{\phi})\right]}{\mathbb{E}\left[T^* \mid \mathsf{RR}(\boldsymbol{\phi})\right]}}_{(d)=\eta_n^{\boldsymbol{\phi}}} \\
&= \sum_{\boldsymbol{\phi}\in\Phi} \beta_{\boldsymbol{\phi}}\, \eta_n^{\boldsymbol{\phi}} \overset{(e)}{>} \lambda_n + \epsilon,
\end{aligned} \tag{3.39}$$

where (a) is by (3.34)(3.36), (b) is by rearranging terms, (c) is by plugging (3.35) into (3.21), (d) is by plugging (3.35) and (3.37) into (3.10) in Section 3.3.1, and (e) is by (3.14). From (3.39), we get

$$\mathbb{E}\left[R_n^*\right] > (\lambda_n + \epsilon)\, \mathbb{E}\left[T^*\right], \quad \text{for all } n \in \{1, \ldots, N\}. \tag{3.40}$$

(**Lyapunov Drift**) From (3.13), in a frame of size T (which is possibly random), we can show that, for all n,

$$Q_n(t+T) \leq \max\left[Q_n(t) - \sum_{\tau=0}^{T-1} \mu_n(t+\tau), 0 \right] + \sum_{\tau=0}^{T-1} a_n(t+\tau). \tag{3.41}$$

We define a Lyapunov function $L(\boldsymbol{Q}(t)) \triangleq (1/2) \sum_{n=1}^{N} Q_n^2(t)$ and the T-slot Lyapunov drift

$$\Delta_T(\boldsymbol{Q}(t)) \triangleq \mathbb{E}\left[L(\boldsymbol{Q}(t+T) - L(\boldsymbol{Q}(t)) \mid \boldsymbol{Q}(t) \right],$$

where, in the last term, the expectation is with respect to the randomness of the whole network in frame T, including the randomness of T. By taking the square of (3.41) and then conditional expectation on $\boldsymbol{Q}(t)$, we can show

$$\begin{aligned}\Delta_T(\boldsymbol{Q}(t)) &\leq \frac{1}{2}N(1+A_{\max}^2)\,\mathbb{E}\left[T^2 \mid \boldsymbol{Q}(t)\right] \\ &- \mathbb{E}\left[\sum_{n=1}^{N} Q_n(t)\left[\sum_{\tau=0}^{T-1}\left(\mu_n(t+\tau)-a_n(t+\tau)\right)\right] \mid \boldsymbol{Q}(t)\right].\end{aligned} \tag{3.42}$$

Define $f(\boldsymbol{Q}(t),\theta)$ as the last term of (3.42), where θ represents a scheduling policy that controls the service rates $\mu_n(t+\tau)$ and the frame size T. In the following analysis, we only consider θ in the class of RandRR policies, and the frame size T is the renewal period of a RandRR policy. By (3.38), the second term of (3.42) is less than or equal to the constant $B_1 \triangleq (1/2)N(1+A_{\max}^2)\mathbb{E}\left[(T_{\max})^2\right] < \infty$. It follows that

$$\Delta_T(\boldsymbol{Q}(t)) \leq B_1 - f(\boldsymbol{Q}(t),\theta). \tag{3.43}$$

In $f(\boldsymbol{Q}(t),\theta)$, it is useful to consider $\theta = \mathsf{RandRR}^*$ and T is the renewal period T^* of RandRR^*. Assume t is the beginning of a renewal period. For each $n \in \{1,2,\ldots,N\}$, because R_n^* is the number of successful transmissions user n receives in the renewal period T^*, we have

$$\mathbb{E}\left[\sum_{\tau=0}^{T^*-1}\mu_n(t+\tau) \mid \boldsymbol{Q}(t)\right] = \mathbb{E}\left[R_n^*\right].$$

Combining it with (3.40), we get

$$\mathbb{E}\left[\sum_{\tau=0}^{T^*-1}\mu_n(t+\tau) \mid \boldsymbol{Q}(t)\right] > (\lambda_n+\epsilon)\,\mathbb{E}\left[T^*\right]. \tag{3.44}$$

By the assumption that packet arrivals are i.i.d. over slots and independent of the current queue backlogs, we have for all n

$$\mathbb{E}\left[\sum_{\tau=0}^{T^*-1}a_n(t+\tau) \mid \boldsymbol{Q}(t)\right] = \lambda_n\,\mathbb{E}\left[T^*\right]. \tag{3.45}$$

Plugging (3.44) and (3.45) into $f(\boldsymbol{Q}(t), \mathsf{RandRR}^*)$, we get

$$f(\boldsymbol{Q}(t), \mathsf{RandRR}^*) \geq \epsilon\, \mathbb{E}\left[T^*\right] \sum_{n=1}^{N} Q_n(t). \tag{3.46}$$

It is also useful to consider θ as a round robin policy $\mathsf{RR}(\boldsymbol{\phi})$ for some $\boldsymbol{\phi} \in \Phi$. Note that $\mathsf{RR}(\boldsymbol{\phi})$ is a special case of RandRR, where we execute $\mathsf{RR}(\boldsymbol{\phi})$ in every round. In this case, the frame size T is the renewal period $T^{\boldsymbol{\phi}}$ of $\mathsf{RR}(\boldsymbol{\phi})$. From Corollary 3.7, we have

$$\mathbb{E}\left[T^{\boldsymbol{\phi}} \mid \boldsymbol{Q}(t)\right] = \mathbb{E}\left[T^{\boldsymbol{\phi}}\right] = \sum_{n:\phi_n=1} \mathbb{E}\left[L_{1n}^{\boldsymbol{\phi}}\right], \tag{3.47}$$

where $\mathbb{E}\left[L_{1n}^{\boldsymbol{\phi}}\right]$ is in (3.9). Let t be the beginning of a transmission round. If channel n is active, we have

$$\mathbb{E}\left[\sum_{\tau=0}^{T^{\boldsymbol{\phi}}-1} \mu_n(t+\tau) \mid \boldsymbol{Q}(t)\right] = \mathbb{E}\left[L_{1n}^{\boldsymbol{\phi}}\right] - 1,$$

and 0 otherwise. It follows that

$$\begin{aligned} f(\boldsymbol{Q}(t), \mathsf{RR}(\boldsymbol{\phi})) &= \left(\sum_{n:\phi_n=1} Q_n(t)\mathbb{E}\left[L_{1n}^{\boldsymbol{\phi}} - 1\right]\right) - \mathbb{E}\left[T^{\boldsymbol{\phi}}\right] \sum_{n=1}^{N} Q_n(t)\lambda_n \\ &\overset{(a)}{=} \sum_{n:\phi_n=1} \left[Q_n(t)\mathbb{E}\left[L_{1n}^{\boldsymbol{\phi}} - 1\right] - \mathbb{E}\left[L_{1n}^{\boldsymbol{\phi}}\right] \sum_{n=1}^{N} Q_n(t)\lambda_n\right], \end{aligned} \tag{3.48}$$

where (a) is by (3.47) and rearranging terms.

(**Design of** QRR) Given the current queue backlogs $\boldsymbol{Q}(t)$, we are interested in the policy that maximizes $f(\boldsymbol{Q}(t), \theta)$ over all RandRR policies in one round of transmission. Although the space of RandRR policies is uncountably large, next we show that the desired policy is a round robin policy $\mathsf{RR}(\boldsymbol{\phi})$ for some $\boldsymbol{\phi} \in \Phi$, and can be found by maximizing $f(\boldsymbol{Q}(t), \mathsf{RR}(\boldsymbol{\phi}))$ in (3.48) over $\boldsymbol{\phi} \in \Phi$. To see this, we denote by $\boldsymbol{\phi}(t)$ the binary vector that maximizes $f(\boldsymbol{Q}(t), \mathsf{RR}(\boldsymbol{\phi}))$ over $\boldsymbol{\phi} \in \Phi$. We have

$$f(\boldsymbol{Q}(t), \mathsf{RR}(\boldsymbol{\phi}(t))) \geq f(\boldsymbol{Q}(t), \mathsf{RR}(\boldsymbol{\phi})), \text{ for all } \boldsymbol{\phi} \in \Phi. \tag{3.49}$$

For any RandRR policy, conditioning on the policy $\mathsf{RR}(\boldsymbol{\phi})$ chosen for the current round of scheduling, we have

$$f(\boldsymbol{Q}(t), \mathsf{RandRR}) = \sum_{\boldsymbol{\phi} \in \Phi} \alpha_{\boldsymbol{\phi}} f(\boldsymbol{Q}(t), \mathsf{RR}(\boldsymbol{\phi})), \tag{3.50}$$

where $\{\alpha_{\boldsymbol{\phi}}\}_{\boldsymbol{\phi} \in \Phi}$ is the probability distribution associated with RandRR. By (3.49)(3.50), for any RandRR we get

$$\begin{aligned} f(\boldsymbol{Q}(t), \mathsf{RR}(\boldsymbol{\phi}(t))) &\geq \sum_{\boldsymbol{\phi} \in \Phi} \alpha_{\boldsymbol{\phi}} f(\boldsymbol{Q}(t), \mathsf{RR}(\boldsymbol{\phi})) \\ &= f(\boldsymbol{Q}(t), \mathsf{RandRR}). \end{aligned} \tag{3.51}$$

We note that, as long as the queue backlog vector $\boldsymbol{Q}(t)$ is not identically zero and the arrival rate vector $\boldsymbol{\lambda}$ is strictly within the inner capacity bound Λ_{int}, we get

$$\begin{aligned} \max_{\boldsymbol{\phi} \in \Phi} f(\boldsymbol{Q}(t), \mathsf{RR}(\boldsymbol{\phi})) &\overset{(a)}{=} f(\boldsymbol{Q}(t), \mathsf{RR}(\boldsymbol{\phi}(t))) \\ &\overset{(b)}{\geq} f(\boldsymbol{Q}(t), \mathsf{RandRR}^*) \\ &\overset{(c)}{>} 0, \end{aligned} \tag{3.52}$$

where (a) is from the definition of $\boldsymbol{\phi}(t)$, (b) from (3.51), and (c) from (3.46).

The policy QRR is designed to be a frame-based algorithm where, at the beginning of each round, we observe the current queue backlog vector $\boldsymbol{Q}(t)$, find the binary vector $\boldsymbol{\phi}(t)$ whose associated round robin policy $\mathsf{RR}(\boldsymbol{\phi}(t))$ maximizes $f(\boldsymbol{Q}(t), \mathsf{RandRR})$ over RandRR policies, and execute $\mathsf{RR}(\boldsymbol{\phi}(t))$ for one round of transmission. We emphasize that, in every transmission round of QRR, active channels are served in the order that the least recently used channel is served first, and the ordering may change from one round to another.

(**Stability Analysis**) Again, policy QRR comprises of a sequence of transmission rounds, where, in each round, QRR finds and executes policy $\mathsf{RR}(\boldsymbol{\phi}(t))$ for one round.

Different $\boldsymbol{\phi}(t)$ may be used in different rounds. In the kth round, let T_k^{QRR} denote its time duration. Define $t_k = \sum_{i=1}^k T_i^{\mathsf{QRR}}$ for all $k \in \mathbb{N}$ and note that $t_k - t_{k-1} = T_k^{\mathsf{QRR}}$. Let $t_0 = 0$. Then, for each $k \in \mathbb{N}$, from (3.43) we have

$$\begin{aligned}\Delta_{T_k^{\mathsf{QRR}}}(\boldsymbol{Q}(t_{k-1})) &\overset{(a)}{\leq} B_1 - f(\boldsymbol{Q}(t_{k-1}), \mathsf{QRR}) \\ &\overset{(b)}{\leq} B_1 - f(\boldsymbol{Q}(t_{k-1}), \mathsf{RandRR}^*) \\ &\overset{(c)}{\leq} B_1 - \epsilon\, \mathbb{E}\left[T^*\right] \sum_{n=1}^{N} Q_n(t_{k-1}),\end{aligned} \tag{3.53}$$

where (a) is by (3.43), (b) is because QRR is the maximizer of $f(\boldsymbol{Q}(t_{k-1}), \mathsf{RandRR})$ over all RandRR policies, and (c) is by (3.46). By taking expectation over $\boldsymbol{Q}(t_{k-1})$ in (3.53) and noting that $\mathbb{E}\left[T^*\right] \geq 1$, for all $k \in \mathbb{N}$ we get

$$\begin{aligned}\mathbb{E}\left[L(\boldsymbol{Q}(t_k))\right] - \mathbb{E}\left[L(\boldsymbol{Q}(t_{k-1}))\right] &\leq B_1 - \epsilon\, \mathbb{E}\left[T^*\right] \sum_{n=1}^{N} \mathbb{E}\left[Q_n(t_{k-1})\right] \\ &\leq B_1 - \epsilon \sum_{n=1}^{N} \mathbb{E}\left[Q_n(t_{k-1})\right].\end{aligned} \tag{3.54}$$

Summing (3.54) over $k \in \{1, 2, \ldots, K\}$, we have

$$\mathbb{E}\left[L(\boldsymbol{Q}(t_K))\right] - \mathbb{E}\left[L(\boldsymbol{Q}(t_0))\right] \leq K B_1 - \epsilon \sum_{k=1}^{K} \sum_{n=1}^{N} \mathbb{E}\left[Q_n(t_{k-1})\right].$$

Since $\boldsymbol{Q}(t_K) \geq \mathbf{0}$ entrywise and by assumption $\boldsymbol{Q}(t_0) = \boldsymbol{Q}(0) = \mathbf{0}$, we have

$$\epsilon \sum_{k=1}^{K} \sum_{n=1}^{N} \mathbb{E}\left[Q_n(t_{k-1})\right] \leq K B_1. \tag{3.55}$$

Dividing (3.55) by ϵK and passing $K \to \infty$, we get

$$\limsup_{K \to \infty} \frac{1}{K} \sum_{k=1}^{K} \sum_{n=1}^{N} \mathbb{E}\left[Q_n(t_{k-1})\right] \leq \frac{B_1}{\epsilon} < \infty. \tag{3.56}$$

Inequality (3.56) shows that the network is stable when sampled at renewal time instants $\{t_k\}$. Then, that it is also stable when sampled over all time follows because: (1) T_k^{QRR}, the renewal period of the $\mathsf{RR}(\boldsymbol{\phi})$ policy chosen in the kth round of QRR, has finite first and second moments for all k (see (3.38)); (2) in every slot, the number of packet arrivals to a user is bounded. These details are provided in the next Lemma 3.20, proved in Section 3.8.9.

Lemma 3.20. Given that

$$\mathbb{E}\left[T_k^{\mathsf{QRR}}\right] \leq \mathbb{E}\left[T_{\max}\right], \quad \mathbb{E}\left[(T_k^{\mathsf{QRR}})^2\right] \leq \mathbb{E}\left[(T_{\max})^2\right] \tag{3.57}$$

for all $k \in \{1, 2, \ldots\}$, packets arrivals to a user is bounded by $A_{\max}$ in every slot, and the network sampled at renewal epochs $\{t_k\}$ is stable from (3.56), we have

$$\limsup_{K\to\infty} \frac{1}{t_K} \sum_{\tau=0}^{t_K-1} \sum_{n=1}^{N} \mathbb{E}\left[Q_n(\tau)\right] < \infty.$$

3.8.9 Proof of Lemma 3.20

In $[t_{k-1}, t_k)$, it is easy to see that, for all $n \in \{1, \ldots, N\}$,

$$Q_n(t_{k-1} + \tau) \leq Q_n(t_{k-1}) + \tau A_{\max}, \quad 0 \leq \tau < T_k^{\mathsf{QRR}}. \tag{3.58}$$

Summing (3.58) over $\tau \in \{0, 1, \ldots, T_k^{\mathsf{QRR}} - 1\}$, we get

$$\sum_{\tau=0}^{T_k^{\mathsf{QRR}}-1} Q_n(t_{k-1} + \tau) \leq T_k^{\mathsf{QRR}} Q_n(t_{k-1}) + \frac{(T_k^{\mathsf{QRR}})^2 A_{\max}}{2}. \tag{3.59}$$

Summing (3.59) over $k \in \{1, 2, \ldots, K\}$ and noting that $t_K = \sum_{k=1}^{K} T_k^{\mathsf{QRR}}$, we have

$$\begin{aligned}
\sum_{\tau=0}^{t_K-1} Q_n(\tau) &= \sum_{k=1}^{K} \sum_{\tau=0}^{T_k^{\mathsf{QRR}}-1} Q_n(t_{k-1}+\tau) \\
&\overset{(a)}{\leq} \sum_{k=1}^{K} \left[T_k^{\mathsf{QRR}} Q_n(t_{k-1}) + \frac{(T_k^{\mathsf{QRR}})^2 A_{\max}}{2} \right],
\end{aligned} \tag{3.60}$$

where (a) is by (3.59). Taking expectation of (3.60) and dividing it by t_K, we have

$$\begin{aligned}
\frac{1}{t_K} \sum_{\tau=0}^{t_K-1} \mathbb{E}\left[Q_n(\tau)\right] &\overset{(a)}{\leq} \frac{1}{K} \sum_{\tau=0}^{t_K-1} \mathbb{E}\left[Q_n(\tau)\right] \\
&\overset{(b)}{\leq} \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}\left[T_k^{\mathsf{QRR}} Q_n(t_{k-1}) + \frac{(T_k^{\mathsf{QRR}})^2 A_{\max}}{2} \right],
\end{aligned} \tag{3.61}$$

where (a) follows $t_K \geq K$ and (b) is by (3.60). Next, we have

$$\begin{aligned}
\mathbb{E}\left[T_k^{\mathsf{QRR}} Q_n(t_{k-1})\right] &= \mathbb{E}\left[\mathbb{E}\left[T_k^{\mathsf{QRR}} Q_n(t_{k-1}) \mid Q_n(t_{k-1})\right]\right] \\
&\overset{(a)}{\leq} \mathbb{E}\left[\mathbb{E}\left[T_{\max} Q_n(t_{k-1}) \mid Q_n(t_{k-1})\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[T_{\max}\right] Q_n(t_{k-1})\right] \\
&= \mathbb{E}\left[T_{\max}\right] \mathbb{E}\left[Q_n(t_{k-1})\right],
\end{aligned} \tag{3.62}$$

where (a) is because $\mathbb{E}\left[T_k^{\mathsf{QRR}}\right] \leq \mathbb{E}\left[T_{\max}\right]$. Using (3.57)(3.62) to upper bound the last term of (3.61), we have

$$\frac{1}{t_K} \sum_{\tau=0}^{t_K-1} \mathbb{E}\left[Q_n(\tau)\right] \leq B_2 + \mathbb{E}\left[T_{\max}\right] \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}\left[Q_n(t_{k-1})\right], \tag{3.63}$$

where $B_2 \triangleq \frac{1}{2}\mathbb{E}\left[(T_{\max})^2\right] A_{\max} < \infty$. Summing (3.63) over $n \in \{1, \ldots, N\}$ and passing $K \to \infty$, we get

$$\begin{aligned}\limsup_{K\to\infty} \frac{1}{t_K} \sum_{\tau=0}^{t_K-1} \sum_{n=1}^{N} \mathbb{E}\left[Q_n(\tau)\right] &\leq NB_2 + \mathbb{E}\left[T_{\max}\right] \left(\limsup_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} \sum_{n=1}^{N} \mathbb{E}\left[Q_n(t_{k-1})\right]\right) \\ &\overset{(a)}{\leq} NB_2 + \mathbb{E}\left[T_{\max}\right] B_1/\epsilon < \infty,\end{aligned}$$

where (a) is by (3.56). The proof is complete. □

Chapter 4

Throughput Utility Maximization over Markovian Channels

In Chapter 3, we seek to characterize the network capacity region Λ over a set of Markov ON/OFF channels. We consider the scenario where instantaneous channel states are never known, at most one user is blindly served per slot, and all scheduling decisions are based on information provided by ACK/NACK feedback from past transmissions. Computing Λ is difficult because it is a restless multi-armed bandit problem. We settle for constructing a good inner bound Λ_{int} on the network capacity region by randomizing round robin policies. This is motivated by that, in practice, we are interested in an easily achievable throughput region with good properties.

This chapter addresses throughput utility maximization over the network model in Chapter 3. Ideally, we consider the problem

$$\text{maximize:} \quad g(\overline{\boldsymbol{y}}), \quad \text{subject to:} \quad \overline{\boldsymbol{y}} \in \Lambda, \tag{4.1}$$

where $g(\cdot)$ is a utility function of the throughput vector $\overline{\boldsymbol{y}}$. We assume that $g(\cdot)$ is concave, continuous, nonnegative, and nondecreasing. However, since the network capacity region Λ is unknown, solving (4.1) is difficult. In this chapter, we instead solve

$$\text{maximize:} \quad g(\overline{\boldsymbol{y}}), \quad \text{subject to:} \quad \overline{\boldsymbol{y}} \in \Lambda_{\text{int}} \tag{4.2}$$

over the inner capacity region Λ_{int} as an approximation to (4.1). We focus on providing the methodology to solve (4.2). The closeness of (4.2) to the original problem (4.1) relies on the tightness of Λ_{int}. This is discussed in Chapter 3 in the special case of symmetric channels, but the inner performance region Λ_{int} may be gradually improved by a deeper investigation into the problem structure.

Scheduling over Markov ON/OFF channels with unknown states has many applications, such as in:

1. Utility maximization over wireless networks with limited channel probing capability and partially observable channels is an interesting problem. For example, we note that the network model is Chapter 3 is the same as one that explicitly probes and serves one channel per slot; the only difference is how channels are probed: explicit probing by signaling or implicit probing by ACK/NACK feedback.

2. In cognitive radio networks [Zhao and Sadler 2007, Zhao and Swami 2007], a secondary user has access to a collection of Markov ON/OFF channels. Every Markovian channel reflects the occupancy of a spectrum by a primary user. The goal of the secondary user is to opportunistically transmit data over unused spectrums for better spectrum efficiency.

3. In target tracking of unmanned aerial vehicles (UAVs) [Ny et al. 2008], a UAV detects one of the many targets in every slot. A Markovian channel here reflects the movement of a target. A channel is ON if its associated target moves to a hotspot, and OFF otherwise. Delivering a packet over a channel represents gaining a reward by locating a target at its hotspot.

4. The problem (4.2) provides an achievable region approach to optimize nonlinear objectives and average reward vectors in a restless multi-armed bandit system.

In these applications, we may be concerned with maximizing weighted sum rewards or providing some notion of fairness across users, spectrums, or targets. In the former, we can let $g(\overline{\boldsymbol{y}}) = \sum_{n=1}^{N} c_n \, \overline{y}_n$, where c_n are positive constants. In the latter, an example is to provide a variant of rate proportional fairness [Kelly 1997, Neely et al. 2008] by using

$$g(\overline{\boldsymbol{y}}) = \sum_{n=1}^{N} c_n \log(1 + \overline{y}_n).$$

The problem (4.2) cannot be solved using existing MaxWeight approaches. The main contribution of this chapter is to provide a new *ratio MaxWeight* method that solves (4.2). More generally, this new ratio rule is suitable to solve utility maximization problems over frame-based systems that have policy-dependent random frame sizes (it will be used in Chapter 5). The resulting network control policy that solves (4.2) has two components. First, a dynamic queue-dependent round robin policy that schedules users for data transmission is constructed. Second, to facilitate the solution to (4.2), an admission control policy is designed. We assume that the network is overloaded by excessive traffic, and the amount of data admitted in every slot is decided by solving a simple convex program.[1] We show that the admission control and channel scheduling policy yields a throughput vector $\overline{\boldsymbol{y}}$ satisfying

$$g(\overline{\boldsymbol{y}}) \geq g(\overline{\boldsymbol{y}}^*) - \frac{B}{V}, \tag{4.3}$$

where $g(\overline{\boldsymbol{y}}^*)$ is the optimal objective of problem (4.2), $B > 0$ is a finite constant, $V > 0$ is a predefined control parameter, and we temporarily assume that all limits exist. By choosing V sufficiently large, the throughput utility $g(\overline{\boldsymbol{y}})$ can be made arbitrarily close to the optimal $g(\overline{\boldsymbol{y}}^*)$, at the expense of increasing average queue sizes. The proof of (4.3) does not require the knowledge of the optimal utility $g(\overline{\boldsymbol{y}}^*)$.

The rest of the chapter is organized as follows. The detailed network model is in Section 4.1. Section 4.2 introduces useful properties of the randomized round robin policies RandRR that are used to construct the inner performance region Λ_{int} in Chapter 3. Our

[1]The admission control decision decouples into separable one-dimensional convex programs that are easily solved in real time when the throughput utility $g(\overline{\boldsymbol{y}})$ is a sum of one-dimensional utility functions.

dynamic control policy is motivated and given in Section 4.3, followed by performance analysis in Section 4.4.

4.1 Network Model

We will use the network model in Chapter 3 with the following additional setup. We suppose that every user has a higher-layer data source of unlimited packets at the base station. The base station keeps a network-layer queue $Q_n(t)$ of infinite capacity for every user $n \in \{1, \dots, N\}$, where $Q_n(t)$ denotes the backlog for user n in slot t. In every slot, the base station admits $r_n(t) \in [0, 1]$ packets for user n from its data source into queue $Q_n(t)$. For simplicity, we assume that $r_n(t)$ takes real values in $[0, 1]$ for all n,[2] and define $\boldsymbol{r}(t) \triangleq (r_n(t))_{n=1}^N$. Let $\mu_n(t) \in \{0, 1\}$ denote the service rate for user n in slot t. The queueing process $\{Q_n(t)\}_{t=0}^\infty$ of user n evolves as

$$Q_n(t+1) = \max[Q_n(t) - \mu_n(t), 0] + r_n(t). \tag{4.4}$$

Initially $Q_n(0) = 0$ for all n. All queues are (strongly) stable if

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{n=1}^{N} \mathbb{E}\left[Q_n(\tau)\right] < \infty. \tag{4.5}$$

4.2 Randomized Round Robin Policy RandRR

We summarize useful properties of the randomized round robin policies RandRR that are used to construct the inner network capacity region Λ_{int} in Chapter 3.

[2] We can accommodate the integer-value assumption of $r_n(t)$ by introducing *auxiliary queues*; see Neely et al. [2008] for an example.

Randomized Round Robin Policy RandRR**:**

1. In every round, pick a binary vector $\boldsymbol{\phi} \in \Phi \cup \{\mathbf{0}\}$ with some probability $\alpha_{\boldsymbol{\phi}}$, where $\alpha_{\mathbf{0}} + \sum_{\boldsymbol{\phi}\in\Phi} \alpha_{\boldsymbol{\phi}} = 1$.

2. If a vector $\boldsymbol{\phi} \in \Phi$ is selected, run policy $\mathsf{RR}(\boldsymbol{\phi})$ presented in Section 3.3 for one round. Otherwise, $\boldsymbol{\phi} = \mathbf{0}$, idle the system for one slot. At the end of either case, go to Step 1.

The above RandRR policy is slightly different than that in Chapter 3, because it includes the option of idling the system. The feasibility of this RandRR policy can be similarly proved using results in Chapter 3 and the monotonicity of the k-step transition probabilities $\{\mathsf{P}_{n,01}^{(k)}, \mathsf{P}_{n,11}^{(k)}\}$.

The following is a corollary of Corollary 3.7, and is useful in later analysis.

Corollary 4.1.

1. Let T_k denote the duration of the kth transmission round in a RandRR policy. The random variables T_k are i.i.d. for different k with

$$\mathbb{E}\left[T_k\right] = \alpha_{\mathbf{0}} + \sum_{\boldsymbol{\phi}\in\Phi} \alpha_{\boldsymbol{\phi}} \left(\sum_{n:\phi_n=1} \mathbb{E}\left[L_n^{\boldsymbol{\phi}}\right] \right),$$

which is computed by conditioning on the policy chosen in a round.

2. Let $N_{n,k}$ denote the number of packets served for user n in round T_k. For each fixed n, the random variables $N_{n,k}$ are i.i.d. over different k with $\mathbb{E}\left[N_{n,k}\right] = \sum_{\boldsymbol{\phi}:\phi_n=1} \alpha_{\boldsymbol{\phi}} \mathbb{E}\left[L_n^{\boldsymbol{\phi}} - 1\right]$, which is computed by conditioning on the $\mathsf{RR}(\boldsymbol{\phi})$ policy that is chosen in a round.

3. Because $N_{n,k}$ and T_k are i.i.d. over k, the throughput of user n under a RandRR policy is equal to $\mathbb{E}\left[N_{n,k}\right] / \mathbb{E}\left[T_k\right]$.

4.3 Network Utility Maximization

4.3.1 The QRRNUM policy

Recall that our inner throughput region Λ_{int} is a convex hull of of the zero vector and the performance vectors of the subset of RandRR policies that includes those execute a fixed $\mathsf{RR}(\boldsymbol{\phi})$ policy in every round. Therefore, the problem (4.2) is a well-defined convex program. Yet, solving (4.2) is difficult because the performance region Λ_{int} is represented as a convex hull of 2^N vectors. The following admission control and channel scheduling policy solves (4.2) in a dynamic manner with low complexity.

Utility-Optimal Queue-Dependent Round Robin (QRRNUM):

- (Admission control) At the start of every round, observe the current queue backlog $\boldsymbol{Q}(t) = (Q_1(t), \dots, Q_N(t))$ and solve the convex program

$$\text{maximize:} \quad V\, g(\boldsymbol{r}) - \sum_{n=1}^{N} Q_n(t)\, r_n \tag{4.6}$$

$$\text{subject to:} \quad r_n \in [0,1],\ \forall\, n \in \{1, \dots, N\}, \tag{4.7}$$

  where $V > 0$ is a predefined control parameter, and $\boldsymbol{r} \triangleq (r_n)_{n=1}^N$. Let $(r_n^{\mathsf{QRR}})_{n=1}^N$ be the solution to (4.6)-(4.7). In every slot of the current round, admit r_n^{QRR} packets for every user $n \in \{1, \dots, N\}$ into queue $Q_n(t)$.

- (Channel scheduling) At the start of every round, over all nonzero binary vectors $\boldsymbol{\phi} \in \Phi$, let $\boldsymbol{\phi}^{\mathsf{QRR}}$ be the maximizer of the ratio

$$\frac{\sum_{n=1}^N Q_n(t)\, \mathbb{E}\left[L_n^{\boldsymbol{\phi}} - 1\right] \phi_n}{\sum_{n=1}^N \mathbb{E}\left[L_n^{\boldsymbol{\phi}}\right] \phi_n}, \quad \mathbb{E}\left[L_n^{\boldsymbol{\phi}}\right] = 1 + \frac{\mathsf{P}_{n,01}^{(M(\boldsymbol{\phi}))}}{\mathsf{P}_{n,10}}, \tag{4.8}$$

  where $M(\boldsymbol{\phi})$ denotes the number of ones (i.e., active channels) in $\boldsymbol{\phi}$. If the maximum of (4.8) is positive, run policy $\mathsf{RR}(\boldsymbol{\phi}^{\mathsf{QRR}})$ in Section 3.3 for one round using the channel ordering of least recently used first. Otherwise, idle the system for one slot. At the end of either case, start a new round of admission control and channel scheduling.

When the utility function $g(\cdot)$ is a sum of individual utilities, i.e., $g(\boldsymbol{r}(t)) = \sum_{n=1}^{N} g_n(r_n(t))$, problem (4.6)-(4.7) decouples into N one-dimensional convex programs, each of which maximizes $[V g_n(r_n(t)) - Q_n(t) r_n(t)]$ over $r_n(t) \in [0, 1]$, which can be solved efficiently in real time.

The most complex part of the QRRNUM policy is to maximize the ratio (4.8). In the following we present a bisection algorithm [Neely 2010c, Section 7.3.1] that searches for the maximum of (4.8) with exponentially fast speed. This algorithm is motivated by the next lemma.

Lemma 4.2. ([Neely 2010c, Lemma 7.5]) Let $a(\boldsymbol{\phi})$ and $b(\boldsymbol{\phi})$ denote the numerator and denominator of (4.8), respectively. Define

$$\theta^* \triangleq \max_{\boldsymbol{\phi} \in \Phi} \left\{ \frac{a(\boldsymbol{\phi})}{b(\boldsymbol{\phi})} \right\}, \; c(\theta) \triangleq \max_{\boldsymbol{\phi} \in \Phi} \left[a(\boldsymbol{\phi}) - \theta b(\boldsymbol{\phi}) \right].$$

Then:

$$\text{If } \theta < \theta^*, \text{ then } c(\theta) > 0.$$

$$\text{If } \theta > \theta^*, \text{ then } c(\theta) < 0.$$

$$\text{If } \theta = \theta^*, \text{ then } c(\theta) = 0.$$

Intuition from Lemma 4.2: To search for the maximum ratio θ^*, suppose initially we know θ^* lies in a feasible region $[\theta_{\min}, \theta_{\max}]$ for some $\theta_{\min}$ and $\theta_{\max}$. Let us compute the midpoint $\theta_{\text{mid}} = \frac{1}{2}(\theta_{\min} + \theta_{\max})$ and evaluate $c(\theta_{\text{mid}})$. If $c(\theta_{\text{mid}}) > 0$, Lemma 4.2 indicates that $\theta_{\text{mid}} < \theta^*$, and we know θ^* lies in the new region $[\theta_{\text{mid}}, \theta_{\max}]$, whose size is half of the initial $[\theta_{\min}, \theta_{\max}]$. In other words, one such bisection operation reduces the feasible region of the unknown θ^* by half. By iterating the bisection process, we can find θ^* with exponential speed. The bisection algorithm is presented next.

Bisection Algorithm that Maximizes (4.8):

- Initially, define $\theta_{\min} \triangleq 0$ and

$$\theta_{\max} \triangleq \frac{\left(\sum_{n=1}^{N} Q_n(t)\right) \max_{n\in\{1,\dots,N\}} \left\{\frac{\pi_{n,\mathsf{ON}}}{\mathsf{P}_{n,10}}\right\}}{1 + \min_{n\in\{1,\dots,N\}} \left\{\frac{\mathsf{P}_{n,01}}{\mathsf{P}_{n,10}}\right\}}$$

  so that $\theta_{\min} \leq a(\boldsymbol{\phi})/b(\boldsymbol{\phi}) \leq \theta_{\max}$ for all $\boldsymbol{\phi} \in \Phi$. It follows that $\theta^* \in [\theta_{\min}, \theta_{\max}]$.

- Compute $\theta_{\text{mid}} = \frac{1}{2}(\theta_{\min} + \theta_{\max})$ and $c(\theta_{\text{mid}})$. If $c(\theta_{\text{mid}}) = 0$, we have $\theta^* = \theta_{\text{mid}}$ and

$$\boldsymbol{\phi}^{\mathsf{QRR}} = \text{argmax}_{\boldsymbol{\phi}\in\Phi} \left[a(\boldsymbol{\phi}) - \theta^* b(\boldsymbol{\phi})\right].$$

  When $c(\theta_{\text{mid}}) < 0$, update the feasible region of θ^* as $[\theta_{\min}, \theta_{\text{mid}}]$. If $c(\theta_{\text{mid}}) > 0$, update the region as $[\theta_{\text{mid}}, \theta_{\max}]$. In either case, repeat the bisection process.

The value $c(\theta)$ is easily computed by noticing

$$c(\theta) = \max_{k\in\{1,\dots,N\}} \left\{ \max_{\boldsymbol{\phi}\in\Phi_k} \left[a(\boldsymbol{\phi}) - \theta b(\boldsymbol{\phi})\right] \right\}, \tag{4.9}$$

where $\Phi_k \subset \Phi$ denotes the set of binary vectors having k ones. For every $k \in \{1, \dots, N\}$, the inner maximum of (4.9) is equal to

$$\max_{\boldsymbol{\phi}\in\Phi_k} \left\{ \sum_{n=1}^{N} \left[\frac{\mathsf{P}_{n,01}^{(k)}}{\mathsf{P}_{n,10}} (Q_n(t) - \theta) - \theta \right] \phi_n \right\},$$

which is solved by sorting $\left[\frac{\mathsf{P}_{n,01}^{(k)}}{\mathsf{P}_{n,10}} (Q_n(t) - \theta) - \theta \right]$ and deciding the binary variables $(\phi_1, \dots, \phi_N)$ accordingly.

4.3.2 Lyapunov Drift Inequality

The construction of the QRRNUM policy follows a novel Lyapunov drift argument. We start with constructing a frame-based Lyapunov drift inequality over a frame of size T, where T is possibly random but has a finite second moment bounded by a constant C so that $C \geq \mathbb{E}\left[T^2 \mid \boldsymbol{Q}(t)\right]$ for all t and all possible $\boldsymbol{Q}(t)$. Intuition for the inequality is provided later. By iteratively applying (4.4), it is not hard to show

$$Q_n(t+T) \leq \max\left[Q_n(t) - \sum_{\tau=0}^{T-1} \mu_n(t+\tau), 0\right] + \sum_{\tau=0}^{T-1} r_n(t+\tau) \tag{4.10}$$

for every $n \in \{1, \ldots, N\}$. We define the quadratic Lyapunov function $L(\boldsymbol{Q}(t)) \triangleq \frac{1}{2}\sum_{n=1}^{N} Q_n^2(t)$ as a scalar measure of queue sizes $\boldsymbol{Q}(t)$. Define the T-slot Lyapunov drift

$$\Delta_T(\boldsymbol{Q}(t)) \triangleq \mathbb{E}\left[L(\boldsymbol{Q}(t+T)) - L(\boldsymbol{Q}(t)) \mid \boldsymbol{Q}(t)\right]$$

as a conditional expected change of queue sizes across T slots, where the expectation is with respect to the randomness of the network within the T slots, including the randomness of T. By taking the square of (4.10) for every n, using inequalities

$$\max[a-b, 0] \leq a \quad \forall a \geq 0,$$
$$(\max[a-b, 0])^2 \leq (a-b)^2, \quad \mu_n(t) \leq 1, \quad r_n(t) \leq 1,$$

to simplify terms, summing all resulting inequalities, and taking conditional expectation on $\boldsymbol{Q}(t)$, we get

$$\Delta_T(\boldsymbol{Q}(t)) \leq B - \mathbb{E}\left[\sum_{n=1}^{N} Q_n(t)\left[\sum_{\tau=0}^{T-1} \mu_n(t+\tau) - r_n(t+\tau)\right] \mid \boldsymbol{Q}(t)\right] \tag{4.11}$$

where $B \triangleq NC > 0$ is a constant. Subtracting from both sides of (4.11) the weighted sum $V\mathbb{E}\left[\sum_{\tau=0}^{T-1} g(\boldsymbol{r}(t+\tau)) \mid \boldsymbol{Q}(t)\right]$, where $V > 0$ is a predefined control parameter, we get the Lyapunov drift inequality

$$\Delta_T(\boldsymbol{Q}(t)) - V\mathbb{E}\left[\sum_{\tau=0}^{T-1} g(\boldsymbol{r}(t+\tau)) \mid \boldsymbol{Q}(t)\right] \leq B - f(\boldsymbol{Q}(t)) - h(\boldsymbol{Q}(t)), \tag{4.12}$$

where

$$f(\boldsymbol{Q}(t)) \triangleq \sum_{n=1}^{N} Q_n(t)\mathbb{E}\left[\sum_{\tau=0}^{T-1} \mu_n(t+\tau) \mid \boldsymbol{Q}(t)\right], \tag{4.13}$$

$$h(\boldsymbol{Q}(t)) \triangleq \mathbb{E}\left[\sum_{\tau=0}^{T-1}\left[V\, g(\boldsymbol{r}(t+\tau)) - \sum_{n=1}^{N} Q_n(t)\, r_n(t+\tau)\right] \mid \boldsymbol{Q}(t)\right]. \tag{4.14}$$

The inequality (4.12) holds for any scheduling policy over a frame of any size T.

4.3.3 Intuition

The desired network control policy shall stabilize all queues $(Q_1(t), \ldots, Q_N(t))$ and maximize the throughput utility $g(\cdot)$. For queue stability, we want to minimize the Lyapunov drift $\Delta_T(\boldsymbol{Q}(t))$, because it captures the expected growth of queue sizes over a duration of time. On the other hand, to increase throughput utility, we want to admit more data into the system for service, and maximize the expected sum utility $\mathbb{E}\left[\sum_{\tau=0}^{T-1} g(\boldsymbol{r}(t+\tau)) \mid \boldsymbol{Q}(t)\right]$. Minimizing Lyapunov drift and maximizing throughput utility, however, conflict with each other, because queue sizes increase with more data admitted into the system. To capture this tradeoff, it is natural to minimize a weighted difference of Lyapunov drift and throughput utility, which is the left side of (4.12). The tradeoff is controlled by the positive parameter V. Intuitively, a large V value puts more weights on throughput utility, thus throughput utility is improved, at the expense of the growth of queue sizes reflected in $\Delta_T(\boldsymbol{Q}(t))$. The construction of the inequality (4.12) provides a useful bound on the weighted difference of Lyapunov drift and throughput utility.

The QRRNUM policy that we present earlier and construct in the next section uses the above ideas with two modifications. First, it suffices to minimize a bound on the weighted difference of Lyapunov drift and throughput utility, i.e., the right side of (4.12). Second, since the weighted difference of Lyapunov drift and throughput utility in (4.12) is made over a frame of T slots, where the value T is random and depends on the policy implemented within the frame, it is natural to normalize the weighted difference by the average frame size, and we minimize the resulting ratio (see (4.15)).

4.3.4 Construction of QRRNUM

We consider the policy that, at the start of every round, observes the current queue backlog vector $\boldsymbol{Q}(t)$ and maximizes over all feasible policies the average

$$\frac{f(\boldsymbol{Q}(t)) + h(\boldsymbol{Q}(t))}{\mathbb{E}\left[T \mid \boldsymbol{Q}(t)\right]} \tag{4.15}$$

over a frame of size T. Every feasible policy consists of: (1) an admission control policy that admits packets into queues $\boldsymbol{Q}(t)$ for all users in every slot; (2) a randomized round robin policy RandRR for data delivery. The frame size T in (4.15) is the length of one transmission round under the candidate RandRR policy, and its distribution depends on the backlog vector $\boldsymbol{Q}(t)$ via the queue-dependent choice of RandRR. When the feasible policy that maximizes (4.15) is chosen, it is executed for one round of transmission, after which a new policy is chosen based on the updated ratio of (4.15), and so on.

Next, we simplify the maximization of (4.15); the result is the QRRNUM policy. In $h(\boldsymbol{Q}(t))$, the optimal admitted data vector $\boldsymbol{r}(t+\tau)$ in every slot is independent of the frame size T and of the rate allocations $\mu_n(t+\tau)$ in $f(\boldsymbol{Q}(t))$. In addition, it should be the same for all $\tau \in \{0, \dots, T-1\}$, and is the solution to (4.6)-(4.7). These observations lead to the admission control subroutine in the QRRNUM policy.

Let $\Psi^*(\boldsymbol{Q}(t))$ denote the optimal objective of (4.6)-(4.7). Since the optimal admitted data vector is independent of the frame size T, we get $h(\boldsymbol{Q}(t)) = \mathbb{E}\left[T \mid \boldsymbol{Q}(t)\right] \Psi^*(\boldsymbol{Q}(t))$, and (4.15) is equal to

$$\frac{f(\boldsymbol{Q}(t))}{\mathbb{E}\left[T \mid \boldsymbol{Q}(t)\right]} + \Psi^*(\boldsymbol{Q}(t)). \tag{4.16}$$

It indicates that finding the optimal admission control policy is independent of finding the optimal channel scheduling policy that maximizes the first term of (4.16).

Next, we evaluate the first term of (4.16) under a fixed $\textsf{RandRR}$ policy with parameters $\{\alpha_{\boldsymbol{\phi}}\}_{\boldsymbol{\phi}\in\Phi\cup\{\mathbf{0}\}}$. Conditioning on the choice of $\boldsymbol{\phi}$, i.e., the active channels chosen for service in one round, we get

$$f(\boldsymbol{Q}(t)) = \sum_{\boldsymbol{\phi}\in\Phi\cup\{\mathbf{0}\}} \alpha_{\boldsymbol{\phi}}\, f(\boldsymbol{Q}(t), \textsf{RR}(\boldsymbol{\phi})),$$

where $f(\boldsymbol{Q}(t), \textsf{RR}(\boldsymbol{\phi}))$ denotes the term $f(\boldsymbol{Q}(t))$ evaluated under policy $\textsf{RR}(\boldsymbol{\phi})$ with the channel ordering of least recently used first. For convenience, we denote by $\textsf{RR}(\mathbf{0})$ the policy of idling the system for one slot. Similarly, by conditioning we have [3]

$$\mathbb{E}\left[T\right] = \mathbb{E}\left[T \mid \boldsymbol{Q}(t)\right] = \sum_{\boldsymbol{\phi}\in\Phi\cup\{\mathbf{0}\}} \alpha_{\boldsymbol{\phi}}\, \mathbb{E}\left[T_{\textsf{RR}(\boldsymbol{\phi})}\right],$$

where $T_{\textsf{RR}(\boldsymbol{\phi})}$ denotes the duration of one transmission round under the $\textsf{RR}(\boldsymbol{\phi})$ policy. It follows that

$$\frac{f(\boldsymbol{Q}(t))}{\mathbb{E}\left[T \mid \boldsymbol{Q}(t)\right]} = \frac{\sum_{\boldsymbol{\phi}\in\Phi\cup\{\mathbf{0}\}} \alpha_{\boldsymbol{\phi}}\, f(\boldsymbol{Q}(t), \textsf{RR}(\boldsymbol{\phi}))}{\sum_{\boldsymbol{\phi}\in\Phi\cup\{\mathbf{0}\}} \alpha_{\boldsymbol{\phi}}\, \mathbb{E}\left[T_{\textsf{RR}(\boldsymbol{\phi})}\right]}. \tag{4.17}$$

The next lemma shows that there exists a $\textsf{RR}(\boldsymbol{\phi})$ policy maximizing (4.17) over all $\textsf{RandRR}$ policies.[4] Thus, it suffices to focus on the class of $\textsf{RR}(\boldsymbol{\phi})$ policies in every round.

Lemma 4.3 (Proof in Section 4.7.1)**.** We index $\textsf{RR}(\boldsymbol{\phi})$ policies for all $\boldsymbol{\phi}\in\Phi\cup\{\mathbf{0}\}$. For the $\textsf{RR}(\boldsymbol{\phi})$ policy with index k, define

$$f_k \triangleq f(\boldsymbol{Q}(t), \textsf{RR}(\boldsymbol{\phi})), \quad D_k \triangleq \mathbb{E}\left[T_{\textsf{RR}(\boldsymbol{\phi})}\right].$$

Without loss of generality, assume

$$\frac{f_1}{D_1} \geq \frac{f_k}{D_k}, \quad \forall k \in \{2, 3, \ldots, 2^N\}.$$

[3]Given a fixed policy $\textsf{RandRR}$, the frame size T no longer depends on the backlog vector $\boldsymbol{Q}(t)$, and $\mathbb{E}\left[T\right] = \mathbb{E}\left[T \mid \boldsymbol{Q}(t)\right]$.

[4]Within one transmission round, a $\textsf{RR}(\boldsymbol{\phi})$ policy is a special case of a $\textsf{RandRR}$ policy.

Then, for any probability distribution $\{\alpha_k\}_{k\in\{1,\dots,2^N\}}$ with $\alpha_k \geq 0$ and $\sum_{k=1}^{2^N} \alpha_k = 1$, we have

$$\frac{f_1}{D_1} \geq \frac{\sum_{k=1}^{2^N} \alpha_k f_k}{\sum_{k=1}^{2^N} \alpha_k D_k}.$$

By Lemma 4.3, next we evaluate the first term of (4.16) under a given $\mathsf{RR}(\boldsymbol{\phi})$ policy. If $\boldsymbol{\phi} = \mathbf{0}$, we get $f(\boldsymbol{Q}(t))/\mathbb{E}[T \mid \boldsymbol{Q}(t)] = 0$. Otherwise, a $\mathsf{RR}(\boldsymbol{\phi})$ policy, $\boldsymbol{\phi} \in \Phi$, yields

$$\mathbb{E}[T] = \mathbb{E}[T \mid \boldsymbol{Q}(t)] = \sum_{n:\phi_n=1} \mathbb{E}\left[L_n^{\boldsymbol{\phi}}\right],$$

$$\mathbb{E}\left[\sum_{\tau=0}^{T-1} \mu_n(t+\tau) \mid \boldsymbol{Q}(t)\right] = \begin{cases} \mathbb{E}\left[L_n^{\boldsymbol{\phi}}\right] - 1 & \text{if } \phi_n = 1 \\ 0 & \text{if } \phi_n = 0 \end{cases}$$

As a result,

$$\frac{f(\boldsymbol{Q}(t))}{\mathbb{E}[T \mid \boldsymbol{Q}(t)]} = \frac{\sum_{n=1}^{N} Q_n(t)\mathbb{E}\left[L_n^{\boldsymbol{\phi}} - 1\right]\phi_n}{\sum_{n=1}^{N} \mathbb{E}\left[L_n^{\boldsymbol{\phi}}\right]\phi_n}. \tag{4.18}$$

The above simplifications lead to the channel scheduling subroutine of the QRRNUM policy.

4.4 Performance Analysis

Theorem 4.4 (Proof in Section 4.7.2)**.** Let $y_n(t) = \min[Q_n(t), \mu_n(t)]$ be the number of packets delivered to user n in slot t; define $\boldsymbol{y}(t) \triangleq (y_n(t))_{n=1}^N$. Define constant $B \triangleq N\mathbb{E}\left[T_{\max}^2\right]$. For any control parameter $V > 0$, the QRRNUM policy stabilizes all queues $\{Q_1(t), \dots, Q_N(t)\}$ and yields throughput utility satisfying

$$\liminf_{t\to\infty} g\left(\frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}[\boldsymbol{y}(\tau)]\right) \geq g(\overline{\boldsymbol{y}}^*) - \frac{B}{V}, \tag{4.19}$$

where $g(\overline{\boldsymbol{y}}^*)$ is the solution to the constrained problem (4.2). By taking V sufficiently large, the QRRNUM policy yields performance arbitrarily close to the optimal $g(\overline{\boldsymbol{y}}^*)$, and solves (4.2). The tradeoff of choosing large V values is the linear increase of average queue sizes with V (cf. (4.40)).

4.5 Chapter Summary and Discussions

We provide a theoretical framework to solve throughput utility maximization over partially observable Markov ON/OFF channels. The performance and control are constrained by limiting channel probing capability and delayed/uncertain channel state information, but can be improved by exploiting channel memory. Overall, attacking such problems requires solving (at least approximately) high-dimensional restless multi-armed bandit problems with nonlinear objective functions of time average vector rewards, which are difficult to solve by existing tools such as Whittle's index or Markov decision theory. This chapter provides a new achievable region method. With an inner performance region constructed in Chapter 3, we provide a novel frame-based Lyapunov drift argument that solves the problem with provably near-optimal performance in the inner performance region. We generalize the existing MaxWeight policies to frame-based systems with random frame sizes, and show that a ratio MaxWeight policy is useful for utility maximization problems.

This chapter provides a new mathematical programming method for optimizing nonlinear objective functions of time average reward vectors in restless multi-armed bandit (RMAB) problems. In the literature, RMAB problems are mostly studied with linear objective functions. The two popular methods for linear RMABs, namely Whittle's index [Whittle 1988] and (partially observable) Markov decision theory [Bertsekas 2005], seem difficult to apply to nonlinear RMABs because they are based on dynamic programming ideas. Extensions of our new method to other RMAB problems are interesting future research.

4.6 Bibliographic Notes

In the literature, stochastic utility maximization over queueing networks is solved in [Eryilmaz and Srikant 2006, 2007, Neely 2003, Neely et al. 2008, Stolyar 2005], assuming that channel (network) states are known perfectly and instantly. Limited channel probing in wireless networks is studied in different contexts in Chang and Liu [2007], Chaporkar and Proutiere [2008], Chaporkar et al. [2010, 2009], Guha et al. [2006b], Li and Neely [2010a], where channels are assumed i.i.d. over slots. This chapter generalizes the utility maximization framework in Neely et al. [2008] to wireless networks with limited channel probing capability and time-correlated channels, and uses channel memory to improve performance.

4.7 Proofs in Chapter 4

4.7.1 Proof of Lemma 4.3

Fact 1: Let $\{a_1, a_2, b_1, b_2\}$ be four positive numbers, and suppose there is a bound z such that $a_1/b_1 \leq z$ and $a_2/b_2 \leq z$. Then for any probability θ (where $0 \leq \theta \leq 1$), we have:

$$\frac{\theta a_1 + (1-\theta)a_2}{\theta b_1 + (1-\theta)b_2} \leq z. \tag{4.20}$$

We prove Lemma 4.3 by induction and (4.20). Initially, for any α_1, $\alpha_2 \geq 0$, $\alpha_1 + \alpha_2 = 1$, from $f_1/D_1 \geq f_2/D_2$ we get

$$\frac{f_1}{D_1} \geq \frac{\alpha_1 f_1 + \alpha_2 f_2}{\alpha_1 D_1 + \alpha_2 D_2}.$$

For some $K > 2$, assume

$$\frac{f_1}{D_1} \geq \frac{\sum_{k=1}^{K-1} \alpha_k f_k}{\sum_{k=1}^{K-1} \alpha_k D_k} \tag{4.21}$$

holds for any probability distribution $\{\alpha_k\}_{k=1}^{K-1}$. It follows that, for any probability distribution $\{\alpha_k\}_{k=1}^{K}$, we get

$$\frac{\sum_{k=1}^{K}\alpha_k f_k}{\sum_{k=1}^{K}\alpha_k D_k} = \frac{(1-\alpha_K)\left[\sum_{k=1}^{K-1}\frac{\alpha_k}{1-\alpha_K}f_k\right]+\alpha_K f_K}{(1-\alpha_K)\left[\sum_{k=1}^{K-1}\frac{\alpha_k}{1-\alpha_K}D_k\right]+\alpha_K D_K} \overset{(a)}{\leq} \frac{f_1}{D_1}$$

where (a) follows Fact 1 because $f_1/D_1 \geq f_K/D_K$ by definition and

$$\frac{f_1}{D_1} \geq \frac{\sum_{k=1}^{K-1}\frac{\alpha_k}{1-\alpha_K}f_k}{\sum_{k=1}^{K-1}\frac{\alpha_k}{1-\alpha_K}D_k}$$

by the induction assumption. □

4.7.2 Proof of Theorem 4.4

In the QRRNUM policy, let t_{k-1} and T_k be the beginning and the duration of the kth transmission round, respectively. We have $T_k = t_k - t_{k-1}$ and $t_k = \sum_{i=1}^{k} T_i$ for all $k \in \mathbb{N}$. Assume $t_0 = 0$. Every T_k is the length of a transmission round of some $\mathsf{RR}(\boldsymbol{\phi})$ policy. Define $T_{\max}$ as the length of a transmission round of the policy $\mathsf{RR}(\mathbf{1})$ that serves all channels in every round. For each $k \in \mathbb{N}$, we have that $T_{\max}$ and $T_{\max}^2$ are stochastically larger than T_k and T_k^2, respectively. As a result, we have for all $k \in \mathbb{N}$

$$\mathbb{E}\left[T_k\right] \leq \mathbb{E}\left[T_{\max}\right] < \infty, \quad \mathbb{E}\left[T_k^2\right] \leq \mathbb{E}\left[T_{\max}^2\right] < \infty. \tag{4.22}$$

To analyze the performance of the QRRNUM policy, we compare it to a near-optimal feasible solution. We will adopt the approach in Neely et al. [2008] but generalize it to a frame-based analysis. For some $\epsilon > 0$, consider the *ϵ-constrained* version of problem (4.2):

$$\text{maximize:} \quad g(\overline{\boldsymbol{y}}), \quad \text{subject to:} \quad \overline{\boldsymbol{y}} \in \Lambda_{\text{int}}(\epsilon), \tag{4.23}$$

where $\Lambda_{\text{int}}(\epsilon)$ is the achievable region Λ_{int} stripping an "ϵ-layer" off the boundary:

$$\Lambda_{\text{int}}(\epsilon) \triangleq \{\overline{\boldsymbol{y}} \mid \overline{\boldsymbol{y}} + \epsilon\,\mathbf{1} \in \Lambda_{\text{int}}\},$$

where $\mathbf{1}$ is an all-one vector. Notice that $\Lambda_{\text{int}}(\epsilon) \to \Lambda_{\text{int}}$ as $\epsilon \to 0$. Let $\overline{\boldsymbol{y}}^*(\epsilon) = (\overline{y}_n^*(\epsilon))_{n=1}^N$ be the solution to (4.23), and $\overline{\boldsymbol{y}}^* = (\overline{y}_n^*)_{n=1}^N$ be the solution to problem (4.2). For simplicity, we assume $\overline{\boldsymbol{y}}^*(\epsilon) \to \overline{\boldsymbol{y}}^*$ as $\epsilon \to 0$.[5]

Since the inner throughput region Λ_{int} corresponds to the set of RandRR policies, there exists a randomized round robin policy $\mathsf{RandRR}_\epsilon^*$ that yields the throughput vector $\overline{\boldsymbol{y}}^*(\epsilon) + \epsilon\,\mathbf{1}$. Let T_ϵ^* denote the length of one transmission round under $\mathsf{RandRR}_\epsilon^*$. From Corollary 4.1, we have for every user $n \in \{1, \dots, N\}$:

$$\mathbb{E}\left[\sum_{\tau=0}^{T_\epsilon^*-1} \mu_n(t+\tau) \mid \boldsymbol{Q}(t)\right] = \mathbb{E}\left[\sum_{\tau=0}^{T_\epsilon^*-1} \mu_n(t+\tau)\right] = (\overline{y}_n^*(\epsilon) + \epsilon)\,\mathbb{E}\left[T_\epsilon^*\right]. \tag{4.24}$$

Combining policy $\mathsf{RandRR}_\epsilon^*$ with the admission policy σ^* that sets $r_n(t+\tau) = \overline{y}_n^*(\epsilon)$ for all users n and $\tau \in \{0, \dots, T_\epsilon^* - 1\}$,[6] we get

$$f_\epsilon^*(\boldsymbol{Q}(t)) = \mathbb{E}\left[T_\epsilon^*\right] \sum_{n=1}^N Q_n(t)(\overline{y}_n^*(\epsilon) + \epsilon) \tag{4.25}$$

$$h_\epsilon^*(\boldsymbol{Q}(t)) = \mathbb{E}\left[T_\epsilon^*\right] \left[V\,g(\overline{\boldsymbol{y}}^*(\epsilon)) - \sum_{n=1}^N Q_n(t)\,\overline{y}_n^*(\epsilon)\right] \tag{4.26}$$

where (4.25) and (4.26) are $f(\boldsymbol{Q}(t))$ and $h(\boldsymbol{Q}(t))$ (see (4.13), (4.14)) evaluated under policies $\mathsf{RandRR}_\epsilon^*$ and σ^*, respectively.

---

[5] This property is proved in a similar case in [Neely 2003, Section 5.5.2].

[6] The throughput $\overline{y}_n^*(\epsilon)$ is less than or equal to one. Thus it is a feasible choice of $r_n(t+\tau)$.

Since the QRRNUM policy maximizes (4.15), computing (4.15) under QRRNUM and the joint policy $(\mathsf{RandRR}_{\epsilon}^{*}, \sigma^{*})$ yields

$$
\begin{aligned}
& f_{\mathsf{QRRNUM}}(\boldsymbol{Q}(t_k)) + h_{\mathsf{QRRNUM}}(\boldsymbol{Q}(t_k)) \\
& \geq \mathbb{E}\left[T_{k+1} \mid \boldsymbol{Q}(t_k)\right] \frac{f_{\epsilon}^{*}(\boldsymbol{Q}(t_k)) + h_{\epsilon}^{*}(\boldsymbol{Q}(t_k))}{\mathbb{E}\left[T_{\epsilon}^{*}\right]} \\
& \overset{(a)}{=} \mathbb{E}\left[T_{k+1} \mid \boldsymbol{Q}(t_k)\right] \left[V\, g(\overline{\boldsymbol{y}}^{*}(\epsilon)) + \epsilon \sum_{n=1}^{N} Q_n(t_k)\right] \\
& = \mathbb{E}\left[T_{k+1}\left(V\, g(\overline{\boldsymbol{y}}^{*}(\epsilon)) + \epsilon \sum_{n=1}^{N} Q_n(t_k)\right) \mid \boldsymbol{Q}(t_k)\right],
\end{aligned}
\tag{4.27}
$$

where (a) is from (4.25) and (4.26). The inequality (4.12) under the QRRNUM policy in the $(k+1)$th round of transmission yields

$$
\begin{aligned}
& \Delta_{T_{k+1}}(\boldsymbol{Q}(t_k)) - V\mathbb{E}\left[\sum_{\tau=0}^{T_{k+1}-1} g(\boldsymbol{r}(t_k+\tau)) \mid \boldsymbol{Q}(t_k)\right] \\
& \overset{(a)}{\leq} B - f_{\mathsf{QRRNUM}}(\boldsymbol{Q}(t_k)) - h_{\mathsf{QRRNUM}}(\boldsymbol{Q}(t_k)) \\
& \overset{(b)}{\leq} B - \mathbb{E}\left[T_{k+1}\left(V\, g(\overline{\boldsymbol{y}}^{*}(\epsilon)) + \epsilon \sum_{n=1}^{N} Q_n(t_k)\right) \mid \boldsymbol{Q}(t_k)\right],
\end{aligned}
\tag{4.28}
$$

where (a) is the inequality (4.12) under policy QRRNUM, and (b) uses (4.27). Taking expectation over $\boldsymbol{Q}(t_k)$ in (4.28) and summing it over $k \in \{0, \ldots, K-1\}$, we get

$$
\begin{aligned}
& \mathbb{E}\left[L(\boldsymbol{Q}(t_K))\right] - \mathbb{E}\left[L(\boldsymbol{Q}(t_0))\right] - V\mathbb{E}\left[\sum_{\tau=0}^{t_K-1} g(\boldsymbol{r}(\tau))\right] \\
& \leq BK - Vg(\overline{\boldsymbol{y}}^{*}(\epsilon))\, \mathbb{E}\left[t_K\right] - \epsilon\, \mathbb{E}\left[\sum_{k=0}^{K-1}\left(T_{k+1} \sum_{n=1}^{N} Q_n(t_k)\right)\right].
\end{aligned}
\tag{4.29}
$$

Since queue backlogs $(Q_1(\cdot), \dots, Q_N(\cdot))$ and $L(\boldsymbol{Q}(\cdot))$ are all nonnegative, and by assumption $\boldsymbol{Q}(t_0) = \boldsymbol{Q}(0) = \boldsymbol{0}$, ignoring all backlog-related terms in (4.29) yields

$$\begin{aligned} -V\mathbb{E}\left[\sum_{\tau=0}^{t_K-1} g(\boldsymbol{r}(\tau))\right] &\le BK - Vg(\overline{\boldsymbol{y}}^*(\epsilon))\,\mathbb{E}\left[t_K\right] \\ &\overset{(a)}{\le} B\mathbb{E}\left[t_K\right] - Vg(\overline{\boldsymbol{y}}^*(\epsilon))\,\mathbb{E}\left[t_K\right] \end{aligned} \tag{4.30}$$

where (a) uses $t_K = \sum_{k=1}^{K} T_k \ge K$. Dividing (4.30) by V and rearranging terms, we get

$$\mathbb{E}\left[\sum_{\tau=0}^{t_K-1} g(\boldsymbol{r}(\tau))\right] \ge \left(g(\overline{\boldsymbol{y}}^*(\epsilon)) - \frac{B}{V}\right)\mathbb{E}\left[t_K\right]. \tag{4.31}$$

Recall from Section 4.3.2 that B in (4.31) is an unspecified constant satisfying $B \ge N\mathbb{E}\left[T_k^2 \mid \boldsymbol{Q}(t)\right]$ for all $k \in \mathbb{N}$ and all possible $\boldsymbol{Q}(t)$. From (4.22), it suffices to define $B \triangleq N\mathbb{E}\left[T_{\max}^2\right]$.

Under the QRRNUM policy, let $K(t)$ denote the number of transmission rounds ending by time t; we have $t_{K(t)} \le t < t_{K(t)+1}$. It follows that $0 \le t - \mathbb{E}\left[t_{K(t)}\right] \le \mathbb{E}\left[t_{K(t)+1} - t_{K(t)}\right] = \mathbb{E}\left[T_{K(t)+1}\right]$. Dividing the above by t and passing $t \to \infty$, we get

$$\lim_{t\to\infty} \frac{t - \mathbb{E}\left[t_{K(t)}\right]}{t} = 0. \tag{4.32}$$

Next, the expected sum utility over the first t slots satisfies

$$\begin{aligned} &\sum_{\tau=0}^{t-1} \mathbb{E}\left[g(\boldsymbol{r}(\tau))\right] \overset{(a)}{\ge} \mathbb{E}\left[\sum_{\tau=0}^{t_{K(t)}-1} g(\boldsymbol{r}(\tau))\right] \\ &\overset{(b)}{\ge} \left[g(\overline{\boldsymbol{y}}^*(\epsilon)) - \frac{B}{V}\right]\mathbb{E}\left[t_{K(t)}\right] \\ &= \left[g(\overline{\boldsymbol{y}}^*(\epsilon)) - \frac{B}{V}\right] t - \left[g(\overline{\boldsymbol{y}}^*(\epsilon)) - \frac{B}{V}\right]\left(t - \mathbb{E}\left[t_{K(t)}\right]\right), \end{aligned} \tag{4.33}$$

where (a) uses $t \geq t_{K(t)}$ and that $g(\cdot)$ is nonnegative, and (b) follows (4.31). Dividing (4.33) by t, taking a $\liminf$ as $t \to \infty$, and using (4.32), we get

$$\liminf_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left[g(\boldsymbol{r}(\tau))\right] \geq g(\overline{\boldsymbol{y}}^*(\epsilon)) - \frac{B}{V}. \tag{4.34}$$

Using Jensen's inequality and the concavity of $g(\cdot)$, we get

$$\liminf_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left[g(\boldsymbol{r}(\tau))\right] \leq \liminf_{t\to\infty} g\left(\overline{\boldsymbol{r}}^{(t)}\right), \tag{4.35}$$

where we define the average admission data vector:

$$\overline{\boldsymbol{r}}^{(t)} \triangleq (\overline{r}_n^{(t)})_{n=1}^N, \quad \overline{r}_n^{(t)} \triangleq \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left[r_n(\tau)\right]. \tag{4.36}$$

Combining (4.34) and (4.35) yields

$$\liminf_{t\to\infty} g\left(\overline{\boldsymbol{r}}^{(t)}\right) \geq g(\overline{\boldsymbol{y}}^*(\epsilon)) - \frac{B}{V},$$

which holds for any sufficiently small ϵ. Passing $\epsilon \to 0$ yields

$$\liminf_{t\to\infty} g\left(\overline{\boldsymbol{r}}^{(t)}\right) \geq g(\overline{\boldsymbol{y}}^*) - \frac{B}{V}. \tag{4.37}$$

Finally, we show the network is stable and the limiting throughput utility satisfies

$$\liminf_{t\to\infty} g\left(\overline{\boldsymbol{y}}^{(t)}\right) \geq \liminf_{t\to\infty} g\left(\overline{\boldsymbol{r}}^{(t)}\right), \tag{4.38}$$

where $\overline{\boldsymbol{y}}^{(t)} = (\overline{y}_n^{(t)})_{n=1}^N$ and $\overline{y}_n^{(t)} \triangleq \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left[y_n(\tau)\right]$. Then, combining (4.37) and (4.38) finishes the proof. To prove stability, ignoring the first, second, and fifth term in (4.29) yields

$$\begin{aligned}\epsilon\,\mathbb{E}\left[\sum_{k=0}^{K-1}\left(T_{k+1}\sum_{n=1}^{N}Q_n(t_k)\right)\right] &\leq BK + V\mathbb{E}\left[\sum_{\tau=0}^{t_K-1} g(\boldsymbol{r}(\tau))\right] \\ &\overset{(a)}{\leq} K\left(B + VG_{\max}\mathbb{E}\left[T_{\max}\right]\right)\end{aligned} \tag{4.39}$$

where we define $G_{\max} \triangleq g(\mathbf{1}) < \infty$ as the maximum value of $g(\boldsymbol{r}(\tau))$ (since $g(\cdot)$ is nondecreasing), and (a) uses

$$g(\boldsymbol{r}(\tau)) \leq G_{\max}, \quad \mathbb{E}\left[t_K\right] = \sum_{k=1}^{K}\mathbb{E}\left[T_k\right] \leq K\mathbb{E}\left[T_{\max}\right].$$

Dividing (4.39) by $K\epsilon$, taking a lim sup as $K \to \infty$, and using $T_{k+1} \geq 1$, we get

$$\limsup_{K\to\infty}\frac{1}{K}\mathbb{E}\left[\sum_{k=0}^{K-1}\sum_{n=1}^{N}Q_n(t_k)\right] \leq \frac{B + VG_{\max}\mathbb{E}\left[T_{\max}\right]}{\epsilon} < \infty. \tag{4.40}$$

Inequality (4.40) shows that the average backlog is bounded when sampled at time instants $\{t_k\}_{k=0}^{\infty}$. This property is enough to conclude that the average backlog over the whole time horizon is bounded, i.e., inequality (4.5) holds and the network is stable. It is because the length of each transmission round T_k has a finite second moment and the maximal amount of data admitted to each user in every slot is at most 1; see [Li and Neely 2011a, Lemma 13] for a detailed proof.

It remains to show that network stability leads to (4.38). Recall that $y_n(\tau) = \min[Q_n(\tau), \mu_n(\tau)]$ is the number of user-n packets served in slot τ; (4.4) is re-written as

$$Q_n(\tau+1) = Q_n(\tau) - y_n(\tau) + r_n(\tau). \tag{4.41}$$

Summing (4.41) over $\tau \in \{0, \ldots, t-1\}$, taking expectation and dividing it by t, we get

$$\frac{\mathbb{E}\left[Q_n(t)\right]}{t} = \overline{r}_n^{(t)} - \overline{y}_n^{(t)}, \tag{4.42}$$

where $\overline{r}_n^{(t)}$ is defined in (4.36) and $\overline{y}_n^{(t)}$ is defined similarly. From [Neely 2010b, Theorem 4(c)], the stability of $Q_n(t)$ and (4.42) yield that for all $n \in \{1, \ldots, N\}$:

$$\limsup_{t\to\infty} \frac{\mathbb{E}\left[Q_n(t)\right]}{t} = \limsup_{t\to\infty} \left(\overline{r}_n^{(t)} - \overline{y}_n^{(t)}\right) = 0. \tag{4.43}$$

Noting that $g(\overline{\boldsymbol{y}}^{(t)})$ is bounded for all t, there exists a convergent subsequence $\{g(\overline{\boldsymbol{y}}^{(t_i)})\}_{i=1}^{\infty}$ of $\{g(\overline{\boldsymbol{y}}^{(t)})\}_{t=0}^{\infty}$ such that

$$\lim_{i\to\infty} g\left(\overline{\boldsymbol{y}}^{(t_i)}\right) = \liminf_{t\to\infty} g\left(\overline{\boldsymbol{y}}^{(t)}\right). \tag{4.44}$$

Since the time average $\overline{r}_n^{(t)}$ in (4.36) is bounded for all n and t, there exists a subsequence $\{t_k\}_k$ of $\{t_i\}_i$ such that the sequence $\{\overline{\boldsymbol{r}}^{(t_k)}\}_{k=1}^{\infty}$ converges. From (4.43), we get

$$\lim_{k\to\infty} (\overline{r}_n^{(t_k)} - \overline{y}_n^{(t_k)}) \le \limsup_{t\to\infty} \left(\overline{r}_n^{(t)} - \overline{y}_n^{(t)}\right) = 0$$

and thus

$$\lim_{k\to\infty} \overline{r}_n^{(t_k)} \le \lim_{k\to\infty} \overline{y}_n^{(t_k)}, \quad \text{for all } n \in \{1, \ldots, N\}. \tag{4.45}$$

It follows that

$$\begin{aligned}\liminf_{t\to\infty} g\left(\overline{\boldsymbol{y}}^{(t)}\right) &\overset{(a)}{=} \lim_{k\to\infty} g\left(\overline{\boldsymbol{y}}^{(t_k)}\right) \\ &\overset{(b)}{\ge} \lim_{k\to\infty} g\left(\overline{\boldsymbol{r}}^{(t_k)}\right) \ge \liminf_{t\to\infty} g\left(\overline{\boldsymbol{r}}^{(t)}\right),\end{aligned}$$

where (a) is from (4.44), (b) uses (4.45) and that $g(\cdot)$ is continuous and nondecreasing. $\square$

Chapter 5

Delay-Optimal Control in a Multi-Class M/G/1 Queue with Adjustable Service Rates

In the previous three chapters, we use Lyapunov drift theory to develop throughput and power-optimal control algorithms in wireless networks. We have considered convex utility maximization problems, and implicitly incorporated time-average constraints: queue-stable policies guarantee that the arrival rate to a queue must be less than or equal to the service rate. Broadly, the analysis and the design of network control algorithms in previous chapters provide a new toolset to solve stochastic convex optimization problems with time-average constraints. For one such problem, we construct a virtual queueing system tailored to it, so that queue-stable policies in the virtual system are online algorithms that solve the original problem. This chapter is devoted to using this new optimization method to study delay and rate-optimal control in multi-class queueing systems that have applications in computer systems and cloud computing environments, as introduced in Section 1.2.

Specifically, we consider a multi-class $M/G/1$ queue with adjustable service rates. Jobs are categorized into N classes and served in a nonpreemptive fashion. The service rate $\mu(P(t))$ allocated at time t incurs a service cost $P(t)$. We are concerned with the following four delay-aware and rate allocation problems:

1. Designing a policy such that the average queueing delay in class n, denoted by $\overline{W}_n$, satisfies $\overline{W}_n \leq d_n$ for all $n \in \{1, \dots, N\}$, where $d_n > 0$ are given constants. Here we assume a constant service rate and no rate control.

2. Minimizing a separable convex function $\sum_{n=1}^{N} f_n(\overline{W}_n)$ of the average queueing delay vector $(\overline{W}_n)_{n=1}^{N}$ subject to delay constraints $\overline{W}_n \leq d_n$ for all classes n; assuming a constant service rate.

3. With adjustable service rates, minimizing average service cost subject to delay constraints $\overline{W}_n \leq d_n$ for all classes n.

4. With adjustable service rates, minimizing a separable convex delay function $\sum_{n=1}^{N} f_n(\overline{W}_n)$ subject to an average service cost constraint.

These four problems have increasing complexity, and are presented in this order so that the readers can familiarize themselves with the new methodology that is used.

In this chapter, we design frame-based policies that only update controls across busy periods (i.e., over renewal periods) for the four control problems. The reason of considering these *coarse-grained* policies is to simplify the decision space of the queue to achieve complex performance objectives. When the service rate is constant, frame-based policies have no performance loss in mean queueing delay (see Lemma 5.4 later). Updating rate allocations across busy periods, as specified later in Assumption 5.2, leads to tractable analysis using renewal reward theory. We remark that although a simulation in Ansell et al. [1999] suggests that frame-based policies may incur higher variance in performance than policies that base control actions on job occupancy in the queue, it seems difficult to develop job-level scheduling policies for the problems considered in this chapter. Job-level scheduling, often based on Markov decision theory, suffers from the curse of dimensionality (multiple job classes lead to a multi-dimensional state space), and seems difficult to handle nonlinear objective functions with time average constraints.

The rest of the chapter is organized as follows. The detailed queueing model is in Section 5.1, followed by a summary of useful $M/G/1$ properties in Section 5.2. The four delay-aware and rate control problems are solved in Section 5.3-5.6, followed by simulation results in Section 5.7.

5.1 Queueing Model

We only consider *queueing delay*, not *system delay* (queueing plus service), in this chapter. System delay can be easily incorporated because, in a nonpreemptive system, average queueing and system delay differ by a constant: the average service time. We use *delay* and *queueing delay* interchangeably in the rest of the chapter.

Consider a nonpreemptive $M/G/1$ queue serving jobs divided into N classes. Jobs of class $n \in \{1, \ldots, N\}$ arrive as an independent Poisson process with rate λ_n. All job sizes are independent across classes and independent and identically distributed (i.i.d.) within each class. Let S_n be a random variable with distribution the same as the size of a class n job, with $\mathbb{E}[S_n]$ being the mean job size for class n. As a technical detail, we assume that the first four moments of S_n are finite in every class n; the distribution of S_n is otherwise arbitrary. We assume that when a job arrives, we only know its class but not its actual size.[1] The server has instantaneous service rate $\mu(P(t))$, where $P(t)$ is the associated service cost at time t. We assume that $\mu(P)$ is increasing in P, and $\mu(0) = 0$. We consider a frame-based system, in which each frame consists of an idle period and the following busy period. For every $k \in \mathbb{Z}^+$, let t_k be the start of the kth frame; the kth frame is $[t_k, t_{k+1})$. Let $T_k \triangleq t_{k+1} - t_k$ denote the size of frame k. Define $t_0 = 0$ and assume the system is initially empty. Let $A_{n,k}$ be the set of class n arrivals in frame k. For each job $i \in A_{n,k}$, let $W_{n,k}^{(i)}$ be its queueing delay.

Assumption 5.1. We concentrate on the class of scheduling policies that are: (1) work-conserving (the server is never idle when there is still work to do); (2) non-anticipative

[1] Applying results in this chapter to cases where job size information is available is discussed at the end of Section 5.8.

(control decisions depend only on past and current system states/actions); (3) nonpreemptive (every job is served without interruption); (4) independent of actual job sizes (we have assumed that job sizes are unknown upon their arrivals). In addition, jobs in each class are served according to first-in-first-out (FIFO).[2]

Assumption 5.2. As for rate allocations, we assume that, in every busy period $k \in Z^+$, a fixed per-unit-time service cost $P_k \in [P_{\min}, P_{\max}]$, i.e., a fixed service rate $\mu(P_k)$, is allocated for the duration of the period; the decisions are possible random. Zero service rates are allocated in idle periods with zero service cost. We assume that the maximum cost $P_{\max}$ is finite, but sufficiently large to ensure feasibility of the desired delay constraints. The minimum cost $P_{\min}$ is chosen to be large enough so that the queue is stable even if $P_{\min}$ is used for all time. In particular, for stability we need

$$\sum_{n=1}^{N} \lambda_n \frac{\mathbb{E}[S_n]}{\mu(P_{\min})} < 1 \Rightarrow \mu(P_{\min}) > \sum_{n=1}^{N} \lambda_n \mathbb{E}[S_n].$$

5.1.1 Definition of Average Delay

The average delay under policies that we propose later may not have well-defined limits. Thus, inspired by Neely [2010a], we define

$$\overline{W}_n \triangleq \limsup_{K \to \infty} \frac{\mathbb{E}\left[\sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)}\right]}{\mathbb{E}\left[\sum_{k=0}^{K-1} |A_{n,k}|\right]} \tag{5.1}$$

[2] We note that, under a constant service rate, a given scheduling policy, and the assumption that job sizes in each class are i.i.d. and unknown in advance, any other queueing discipline in a class yields the same average queueing delay as FIFO in that class; hence, our results in this chapter apply to a much broader policy space. In each class, since the average finished work in the system (queueing plus service) and the average residue work in service are both independent of queueing disciplines, so is the average unfinished work in the queue in that class. As a result, the average queueing delay in each class n under two different queueing disciplines satisfies

$$\lambda_n \mathbb{E}[X_n] \overline{W}_n^1 = \lambda_n \mathbb{E}[X_n] \overline{W}_n^2 \Rightarrow \overline{W}_n^1 = \overline{W}_n^2,$$

where $\mathbb{E}[X_n]$ denotes the mean service time of class n jobs, and $\overline{W}_n^1$ and $\overline{W}_n^2$ are mean queueing delay in class n under different queueing disciplines.

as the average queueing delay in class $n \in \{1, \ldots, N\}$, where $|A_{n,k}|$ is the number of class n arrivals in frame k. We only consider average delays sampled at frame boundaries for simplicity. To verify (5.1), the running average delay in class n at time t_K is equal to

$$\frac{\sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)}}{\sum_{k=0}^{K-1} |A_{n,k}|} = \frac{\frac{1}{K} \sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)}}{\frac{1}{K} \sum_{k=0}^{K-1} |A_{n,k}|}.$$

We define two averages

$$w_n^{\text{av}} \triangleq \lim_{K \to \infty} \frac{1}{K} \sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)}, \quad a_n^{\text{av}} \triangleq \lim_{K \to \infty} \frac{1}{K} \sum_{k=0}^{K-1} |A_{n,k}| .$$

If both limits w_n^{av} and a_n^{av} exist, the ratio $w_n^{\text{av}}/a_n^{\text{av}}$ is the limiting average delay in class n. In this case, we get

$$\begin{aligned} \overline{W}_n &= \frac{\lim_{K \to \infty} \mathbb{E}\left[\frac{1}{K} \sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)}\right]}{\lim_{K \to \infty} \mathbb{E}\left[\frac{1}{K} \sum_{k=0}^{K-1} |A_{n,k}|\right]} \\ &= \frac{\mathbb{E}\left[\lim_{K \to \infty} \frac{1}{K} \sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)}\right]}{\mathbb{E}\left[\lim_{K \to \infty} \frac{1}{K} \sum_{k=0}^{K-1} |A_{n,k}|\right]} = \frac{w_n^{\text{av}}}{a_n^{\text{av}}}, \end{aligned} \tag{5.2}$$

which shows that $\overline{W}_n$ is indeed the limiting average delay. The definition in (5.1) replaces the limit by lim sup to guarantee the term is well-defined.

Remark 5.3. The second equality in (5.2), where we pass the limits into the expectations, can be proved by a generalized Lebesgue's dominated convergence theorem stated as follows. Let $\{X_n\}_{n=1}^{\infty}$ and $\{Y_n\}_{n=1}^{\infty}$ be two sequences of random variables such that: (1) $0 \leq |X_n| \leq Y_n$ with probability 1 for all n; (2) for some random variables X and Y, $X_n \to X$ and $Y_n \to Y$ with probability 1; (3) $\lim_{n \to \infty} \mathbb{E}[Y_n] = \mathbb{E}[Y] < \infty$. Then $\mathbb{E}[X]$ is finite and $\lim_{n \to \infty} \mathbb{E}[X_n] = \mathbb{E}[X]$. The details are omitted for brevity.

5.2 Preliminaries

This section summarizes useful properties of a nonpreemptive multi-class $M/G/1$ queue. Here we assume a constant service rate $\mu(P)$ with a fixed cost P (this is extended in Section 5.5). Let $X_n \triangleq S_n/\mu(P)$ be the service time of a class n job. Define $\rho_n \triangleq \lambda_n \mathbb{E}[X_n]$. Fix an arrival rate vector $(\lambda_n)_{n=1}^N$ satisfying $\sum_{n=1}^N \rho_n < 1$; the rate vector $(\lambda_n)_{n=1}^N$ is supportable in the queue.

For each $k \in \mathbb{Z}^+$, let I_k and B_k denote the kth idle and busy period, respectively; the frame size $T_k = I_k + B_k$. The distributions of B_k and T_k are fixed under any work-conserving policy when the service rate is constant. It is because the sample path of unfinished work in the system always decreases at the rate of the server and has a jump when a job arrives, regardless of the order jobs are served. Next, since Poisson arrivals are memoryless, we have $\mathbb{E}[I_k] = 1/(\sum_{n=1}^N \lambda_n)$ for all k. For the same reason, the queueing system renews itself at the start of every frame (i.e., at the start of every idle period). As a result, the frame size T_k, busy period B_k, and the number of class n arrivals in a frame, namely $|A_{n,k}|$, are all i.i.d. over k. Using renewal reward theory [Ross 1996] with renewal epochs defined at frame boundaries $\{t_k\}_{k=0}^\infty$, we have for all $k \in \mathbb{Z}^+$:

$$\mathbb{E}[T_k] = \frac{\mathbb{E}[I_k]}{1-\sum_{n=1}^N \rho_n} = \frac{1}{(1-\sum_{n=1}^N \rho_n)\sum_{n=1}^N \lambda_n}, \tag{5.3}$$

$$\mathbb{E}[|A_{n,k}|] = \lambda_n \,\mathbb{E}[T_k], \quad \forall n \in \{1,\dots,N\}. \tag{5.4}$$

It is useful to consider a stationary randomized priority policy π_{rand} that, in every busy period, randomly chooses one strict priority policy for the duration of the busy period according to a stationary distribution; jobs in each class are served according to FIFO. The mean queueing delay $\overline{W}_n$ of class n under the π_{rand} policy satisfies in every frame $k \in \mathbb{Z}^+$:

$$\mathbb{E}\left[\sum_{i\in A_{n,k}} W_{n,k}^{(i)}\right] = \mathbb{E}\left[\int_{t_k}^{t_{k+1}} Q_n(t)\,\mathrm{d}t\right] = \lambda_n \,\overline{W}_n \,\mathbb{E}[T_k], \quad \forall n \in \{1,\dots,N\}, \tag{5.5}$$

where we recall that $W_{n,k}^{(i)}$ represents only the queueing delay (not including service time), and $Q_n(t)$ denotes the number of class n jobs in the queue (not including that in the server) at time t. The first equality of (5.5) is easily verified by a sample-path argument (see Bertsekas and Gallager [1992, Figure 3.1]), and the second equality follows renewal reward theory and Little's Theorem.

Next we present useful properties of the performance region of mean queueing delay in a nonpreemptive multi-class $M/G/1$ queue under scheduling policies defined in Assumption 5.1. For these results we refer readers to Bertsekas and Gallager [1992], Federgruen and Groenevelt [1988b], Yao [2002] for a detailed introduction. Define $x_n \triangleq \rho_n \overline{W}_n$ for every class $n \in \{1, \dots, N\}$, and denote by Ω the performance region of the vector $(x_n)_{n=1}^N$. The set Ω is *(a base of) a polymatroid* with properties: (1) every vertex of Ω is the performance vector of a strict priority policy; (2) the performance vector of every strict priority policy is a vertex of Ω. In other words, there is a one-to-one mapping between vertices of Ω and strict priority policies. Consequently, every feasible vector $(x_n)_{n=1}^N \in \Omega$, or equivalently every feasible mean queueing delay vector $(\overline{W}_n)_{n=1}^N$, is attained by a randomization of strict priority policies. Such randomization can be implemented across busy periods, since the $M/G/1$ queue renews itself at the end of busy periods. This is formalized in the next lemma.

Lemma 5.4 (Proof in Section 5.10.1)**.** In a multi-class $M/G/1$ queue with scheduling policies defined in Assumption 5.1, we define $\mathcal{W} \triangleq \left\{(\overline{W}_n)_{n=1}^N \mid (\rho_n \overline{W}_n)_{n=1}^N \in \Omega\right\}$ as the performance region of the mean queueing delay vector [Federgruen and Groenevelt 1988b]. Let π_{rand} be the policy that randomly chooses one strict priority policy in every busy period according to a stationary distribution. Then we have:

1. The mean queueing delay vector under every π_{rand} policy lies in $\mathcal{W}$.

2. Conversely, every point in $\mathcal{W}$ is the performance vector of a π_{rand} policy.

Optimizing a linear function over a polymatroid is useful in later analysis; the solution is the well known $c\mu$ rule.

Lemma 5.5 (The $c\mu$ rule; Bertsekas and Gallager [1992], Federgruen and Groenevelt [1988b]). In a multi-class $M/G/1$ queue with scheduling policies defined in Assumption 5.1, we define $x_n \triangleq \rho_n \overline{W}_n$ and consider the linear program:

$$\text{minimize:} \quad \sum_{n=1}^{N} c_n \, x_n \tag{5.6}$$

$$\text{subject to:} \quad (x_n)_{n=1}^{N} \in \Omega \tag{5.7}$$

where c_n are nonnegative constants. We assume $\sum_{n=1}^{N} \rho_n < 1$ for stability, and that the service time of a class n job has a finite second moment $\mathbb{E}\left[X_n^2\right] < \infty$ for all classes n. The optimal solution to (5.6)-(5.7) is a strict priority policy that prioritizes job classes in the decreasing order of c_n. That says, if $c_1 \geq c_2 \geq \ldots \geq c_N$, then class 1 gets the highest priority, class 2 gets the second highest priority, and so on. In this case, the optimal average queueing delay $\overline{W}_n^*$ of class n is equal to

$$\overline{W}_n^* = \frac{R}{(1 - \sum_{k=0}^{n-1} \rho_k)(1 - \sum_{k=0}^{n} \rho_k)}, \tag{5.8}$$

where $\rho_0 \triangleq 0$ and $R \triangleq \frac{1}{2} \sum_{n=1}^{N} \lambda_n \mathbb{E}\left[X_n^2\right]$.

5.3 Achieving Delay Constraints

In this section, we design a policy that yields mean queueing delay of class n satisfying a constraint $\overline{W}_n \leq d_n$ for all classes n. We assume a constant service rate and that all delay constraints are feasible.

Our method transforms this problem into a new queue stability problem. We define a discrete-time *virtual delay queue* $\{Z_{n,k}\}_{k=0}^{\infty}$ for each class $n \in \{1, \ldots, N\}$, where the queue backlog $Z_{n,k+1}$ is computed at time t_{k+1} by

$$Z_{n,k+1} = \max\left[Z_{n,k} + \sum_{i \in A_{n,k}} \left(W_{n,k}^{(i)} - d_n\right), 0\right], \quad \forall \, k \in \mathbb{Z}^+. \tag{5.9}$$

In (5.9), the values $W_{n,k}^{(i)}$ are queueing delays of class n jobs served in the previous frame $[t_k, t_{k+1})$. We assume initially $Z_{n,0} = 0$ for all n. The values $W_{n,k}^{(i)}$ and d_n can be viewed as arrivals and services of the queue $Z_{n,k}$, respectively. If queue $Z_{n,k}$ is stable, we know that the average arrival rate to the queue (being the per-frame average sum of class n delays $\sum_{i\in A_{n,k}} W_{n,k}^{(i)}$) is less than or equal to the average service rate (being the value d_n multiplied by the average number of class n arrivals per frame), from which we infer $\overline{W}_n \leq d_n$. This is formalized below.

Definition 5.6. We say queue $Z_{n,k}$ is mean rate stable if $\lim_{K\to\infty} \mathbb{E}\left[Z_{n,K}\right]/K = 0$.

Lemma 5.7. If queue $Z_{n,k}$ is mean rate stable, then $\overline{W}_n \leq d_n$.

Proof (Lemma 5.7)**.** From (5.9), we get

$$Z_{n,k+1} \geq Z_{n,k} - d_n \left|A_{n,k}\right| + \sum_{i\in A_{n,k}} W_{n,k}^{(i)}.$$

Summing the above over $k \in \{0, \dots, K-1\}$, using $Z_{n,0} = 0$, and taking expectation yields

$$\mathbb{E}\left[Z_{n,K}\right] \geq -d_n \mathbb{E}\left[\sum_{k=0}^{K-1} \left|A_{n,k}\right|\right] + \mathbb{E}\left[\sum_{k=0}^{K-1} \sum_{i\in A_{n,k}} W_{n,k}^{(i)}\right].$$

Dividing the above by $\mathbb{E}\left[\sum_{k=0}^{K-1} \left|A_{n,k}\right|\right]$ yields

$$\frac{\mathbb{E}\left[Z_{n,K}\right]}{\mathbb{E}\left[\sum_{k=0}^{K-1} \left|A_{n,k}\right|\right]} \geq \frac{\mathbb{E}\left[\sum_{k=0}^{K-1} \sum_{i\in A_{n,k}} W_{n,k}^{(i)}\right]}{\mathbb{E}\left[\sum_{k=0}^{K-1} \left|A_{n,k}\right|\right]} - d_n.$$

Taking a lim sup as $K \to \infty$ and using definition (5.1) yields

$$\overline{W}_n \leq d_n + \limsup_{K\to\infty} \frac{\mathbb{E}\left[Z_{n,K}\right]}{K} \frac{K}{\mathbb{E}\left[\sum_{k=0}^{K-1} \left|A_{n,k}\right|\right]}.$$

Using $\mathbb{E}[|A_{n,k}|] = \lambda_n \mathbb{E}[T_k] \geq \lambda_n \mathbb{E}[I_k] = \lambda_n \mathbb{E}[I_0]$ and mean rate stability of $Z_{n,k}$, we get

$$\overline{W}_n \leq d_n + \frac{1}{\lambda_n \mathbb{E}[I_0]} \lim_{K\to\infty} \frac{\mathbb{E}[Z_{n,K}]}{K} = d_n. \quad \square$$

5.3.1 Delay Feasible Policy DelayFeas

The next dynamic index policy, DelayFeas, stabilizes all $Z_{n,k}$ queues in the mean rate stable sense and satisfies all delay constraints $\overline{W}_n \leq d_n$ (by Lemma 5.7).

Delay Feasible Policy DelayFeas:

> In every frame $k \in \mathbb{Z}^+$, the queue updates $Z_{n,k}$ by (5.9) at the beginning of the frame, and serves jobs using strict priorities assigned in the decreasing order of $Z_{n,k}$ (the class with the largest $Z_{n,k}$ has the highest priority) for the duration of the frame. Ties are broken arbitrarily.

Intuitively, the value $Z_{n,k}$ tracks the amount of past queueing delays in class n exceeding the desired delay bound d_n (cf. (5.9)). In every frame, the DelayFeas policy gives priorities to job classes that more severely violate their delay constraints. No statistical knowledge of the queue is needed.

5.3.2 Construction of DelayFeas

The DelayFeas policy is designed using Lyapunov drift theory. For some finite constants $\theta_n > 0$, $n \in \{1, \ldots, N\}$, we define the quadratic Lyapunov function

$$L(\boldsymbol{Z}_k) \triangleq \frac{1}{2} \sum_{n=1}^{N} \theta_n Z_{n,k}^2$$

as a scalar measure of the queue size vector $\boldsymbol{Z}_k \triangleq (Z_{n,k})_{n=1}^{N}$. We define the one-frame Lyapunov drift

$$\Delta(\boldsymbol{Z}_k) \triangleq \mathbb{E}[L(\boldsymbol{Z}_{k+1}) - L(\boldsymbol{Z}_k) \mid \boldsymbol{Z}_k]$$

as the conditional expected growth of the queue size measure $L(\boldsymbol{Z}_k)$ over frame k. The intuition is that minimizing the expected queue growth $\Delta(\boldsymbol{Z}_k)$ over every frame suffices to stabilize all $Z_{n,k}$ queues.

It actually suffices to minimize a bound on $\Delta(\boldsymbol{Z}_k)$ in every frame. Taking the square of (5.9) and using $(\max[a, 0])^2 \leq a^2$ yields

$$Z_{n,k+1}^2 \leq \left[Z_{n,k} + \sum_{i \in A_{n,k}} \left(W_{n,k}^{(i)} - d_n \right) \right]^2 . \tag{5.10}$$

Multiplying (5.10) by $\theta_n/2$, summing over $n \in \{1, \ldots, N\}$, and taking conditional expectation on $\boldsymbol{Z}_k$, we get

$$\begin{aligned} \Delta(\boldsymbol{Z}_k) \leq \frac{1}{2} \sum_{n=1}^{N} \theta_n \, \mathbb{E} \left[\left(\sum_{i \in A_{n,k}} \left(W_{n,k}^{(i)} - d_n \right) \right)^2 \mid \boldsymbol{Z}_k \right] \\ + \sum_{n=1}^{N} \theta_n \, Z_{n,k} \, \mathbb{E} \left[\sum_{i \in A_{n,k}} \left(W_{n,k}^{(i)} - d_n \right) \mid \boldsymbol{Z}_k \right] . \end{aligned} \tag{5.11}$$

Lemma 5.19 in Section 5.10.2 shows that the second term of (5.11) is bounded by a finite constant $C > 0$. Therefore, we get

$$\Delta(\boldsymbol{Z}_k) \leq C + \sum_{n=1}^{N} \theta_n \, Z_{n,k} \, \mathbb{E} \left[\sum_{i \in A_{n,k}} \left(W_{n,k}^{(i)} - d_n \right) \mid \boldsymbol{Z}_k \right] . \tag{5.12}$$

Over all (possibly random) scheduling policies defined in Assumption 5.1, we are interested in the one that minimizes the right side of (5.12) in every frame k. To show that this is the $\mathsf{DelayFeas}$ policy, we simplify (5.12). Under a scheduling policy in Assumption 5.1, we have by renewal reward theory that

$$\mathbb{E} \left[\sum_{i \in A_{n,k}} W_{n,k}^{(i)} \mid \boldsymbol{Z}_k \right] = \lambda_n \overline{W}_{n,k} \, \mathbb{E} \left[T_k \right] ,$$

where $\overline{W}_{n,k}$ denotes the average queueing delay in class n if the (random) control in frame k is repeated in every frame. Together with $\mathbb{E}[|A_{n,k}|] = \lambda_n \mathbb{E}[T_k]$ and the genie decisions $\theta_n = \mathbb{E}[X_n]$ for all classes n, (5.12) is re-written as

$$\Delta(\boldsymbol{Z}_k) \leq \left(C - \mathbb{E}[T_k] \sum_{n=1}^{N} Z_{n,k}\, \rho_n\, d_n\right) + \mathbb{E}[T_k] \sum_{n=1}^{N} Z_{n,k}\, \rho_n\, \overline{W}_{n,k}, \tag{5.13}$$

where $\rho_n = \lambda_n \mathbb{E}[X_n]$. Because the service rate is constant in this section, the second term of (5.13) and $\mathbb{E}[T_k]$ are fixed under all work-conserving policies. It follows that our desired policy minimizes

$$\sum_{n=1}^{N} Z_{n,k}\, \rho_n \overline{W}_{n,k}$$

over all feasible mean queueing delay vectors $(\overline{W}_{n,k})_{n=1}^{N}$; this is the DelayFeas policy from the $c\mu$ rule in Lemma 5.5.

Remark 5.8. In the above analysis, the mean service time $\mathbb{E}[X_n]$ as a value of θ_n is only needed in the arguments constructing the DelayFeas policy. The DelayFeas policy itself does not need the knowledge of $\mathbb{E}[X_n]$.

Remark 5.9. The DelayFeas policy is equivalent to minimizing the ratio of the right side of (5.12) over the expected frame size $\mathbb{E}[T_k]$ in every frame k. This is in the same spirit of the ratio MaxWeight rule that we have used for stochastic optimization over frame-based systems in Chapter 4. All control policies later in this chapter use similar ratio MaxWeight policies. In this section, it reduces to minimize the right side of (5.12) since $\mathbb{E}[T_k]$ is fixed. This ratio rule does not reduce in this way in the last two problems when the service rate is adjustable.

5.3.3 Performance of DelayFeas

Theorem 5.10. For every collection of feasible delay bounds $\{d_1, \dots, d_N\}$ under scheduling policies defined in Assumption 5.1, the DelayFeas policy satisfies $\overline{W}_n \leq d_n$ for all classes $n \in \{1, \dots, N\}$.

Proof (Theorem 5.10)**.** By Lemma 5.7, it suffices to show that the DelayFeas policy stabilizes all $Z_{n,k}$ queues in the mean rate stable sense. By Lemma 5.4, there exists a stationary randomized priority policy π^*_{rand} that yields mean queueing delay $\overline{W}^*_n$ satisfying $\overline{W}^*_n \leq d_n$ for all classes n. Since the DelayFeas policy minimizes the last term of (5.13) in every frame (under $\theta_n = \mathbb{E}[X_n]$ for all n), comparing the DelayFeas policy with the π^*_{rand} policy yields, in every frame k,

$$\sum_{n=1}^{N} \theta_n \, Z_{n,k} \, \lambda_n \, \overline{W}^{\mathsf{DelayFeas}}_{n,k} \leq \sum_{n=1}^{N} \theta_n \, Z_{n,k} \, \lambda_n \, \overline{W}^*_n. \tag{5.14}$$

As a result, (5.13) under the DelayFeas policy satisfies

$$\begin{aligned}
\Delta(\boldsymbol{Z}_k) &\leq C + \mathbb{E}[T_k] \sum_{n=1}^{N} \theta_n \, Z_{n,k} \, \lambda_n (\overline{W}^{\mathsf{DelayFeas}}_{n,k} - d_n) \\
&\leq C + \mathbb{E}[T_k] \sum_{n=1}^{N} \theta_n \, Z_{n,k} \, \lambda_n (\overline{W}^*_n - d_n) \\
&\leq C.
\end{aligned}$$

Taking expectation, summing over $k \in \{0, \ldots, K-1\}$, and noting $L(\boldsymbol{Z}_0) = 0$, we get

$$\mathbb{E}[L(\boldsymbol{Z}_K)] = \frac{1}{2} \sum_{n=1}^{N} \theta_n \mathbb{E}\left[Z^2_{n,K}\right] \leq KC.$$

It follows that $\mathbb{E}\left[Z^2_{n,K}\right] \leq 2KC/\theta_n$ for all classes n, and

$$0 \leq \mathbb{E}[Z_{n,K}] \leq \sqrt{\mathbb{E}\left[Z^2_{n,K}\right]} \leq \sqrt{\frac{2KC}{\theta_n}}, \quad \forall \, n \in \{1, \ldots, N\}.$$

Dividing the above by K yields

$$0 \leq \frac{\mathbb{E}[Z_{n,K}]}{K} \leq \sqrt{\frac{2C}{K\theta_n}}, \quad \forall \, n \in \{1, \ldots, N\}. \tag{5.15}$$

Passing $K \to \infty$ proves mean rate stability for all $Z_{n,k}$ queues. □

Remark 5.11. The convergence time of the DelayFeas policy reflects how soon the running average delay in each class n is below the desired bound d_n. From Lemma 5.7, the speed of the ratio $\mathbb{E}\left[Z_{n,K}\right]/K$ approaching zero gives us a good intuition. According to (5.15), we expect that the DelayFeas policy converges with speed $O(1/\sqrt{K})$, where K is the number of passed busy periods. The control algorithms developed for the next three problems have similar convergence time.

5.4 Convex Delay Optimization

Generalizing the above problem, next we solve the convex delay minimization problem:

$$\text{minimize:} \quad \sum_{n=1}^{N} f_n(\overline{W}_n) \tag{5.16}$$

$$\text{subject to:} \quad \overline{W}_n \leq d_n, \quad \forall\, n \in \{1, \dots, N\} \tag{5.17}$$

over the set of all scheduling policies defined in Assumption 5.1. The functions $f_n(\cdot)$ are assumed continuous, convex, nondecreasing, and nonnegative for all classes n. We assume a constant service rate in this section, and that (5.17) are feasible constraints. The goal is to construct a frame-based policy that solves (5.16)-(5.17). Let $(\overline{W}_n^*)_{n=1}^N$ be the optimal solution to (5.16)-(5.17), attained by a stationary randomized priority policy π^*_{rand} (from Lemma 5.4).

5.4.1 Delay Proportional Fairness

One interesting delay penalty function $f_n(\cdot)$ is the one that attains *proportional fairness*. We say a delay vector $(\overline{W}_n^*)_{n=1}^N$ is *delay proportional fair* if it is optimal under quadratic penalty functions $f_n(\overline{W}_n) = \frac{1}{2}\, c_n (\overline{W}_n)^2$ for all classes n, where $c_n > 0$ are given constants. In this case, any feasible delay vector $(\overline{W}_n)_{n=1}^N$ necessarily satisfies [Bertsekas 1999]

$$\sum_{n=1}^{N} f_n'(\overline{W}_n^*)(\overline{W}_n - \overline{W}_n^*) = \sum_{n=1} c_n(\overline{W}_n - \overline{W}_n^*)\overline{W}_n^* \geq 0, \tag{5.18}$$

which is in the same spirit as the *rate proportional fair* [Kelly 1997] criterion

$$\sum_{n=1}^{N} c_n \frac{x_n - x_n^*}{x_n^*} \leq 0, \tag{5.19}$$

where $(x_n)_{n=1}^N$ is a feasible rate vector and $(x_n^*)_{n=1}^N$ the optimal rate vector. Intuitively, while we favor large rates in rate proportional fairness, small delays (i.e., large inverse of delay) are favored in delay proportional fairness. This suggests that delay proportional fairness shall have the product form (5.18) instead of the ratio form (5.19). To further clarify, we give a two-user example showing that the two criteria (5.18) and (5.19) provide the same proportional tradeoff. Assume $c_1 = c_2 = 1$. In rate proportional fairness, consider two feasible rates $x_1 = 100$ and $x_2 = 10$; user 2 is 10 times worse than user 1. Then, if user 1 wants to increase x units of rate, it cannot cause user 2 more than $x/10$ units of loss. In delay proportional fairness, we suppose $\overline{W}_1 = 10$ and $\overline{W}_2 = 100$; user 2 is again 10 times worse than user 1. Then, according to (5.18), if user 1 wants to decrease delay by x units, user 2 can only tolerate up to $x/10$ units of delay increase.

5.4.2 Delay Fairness Policy DelayFair

Next we present the dynamic index policy that solves (5.16)-(5.17). First we setup virtual queues. In addition to having the $Z_{n,k}$ queues evolving as (5.9) for all classes n, we have new discrete-time virtual queues $\{Y_{n,k}\}_{k=0}^{\infty}$ for all n, where $Y_{n,k+1}$ is computed at time t_{k+1} by

$$Y_{n,k+1} = \max\left[Y_{n,k} + \sum_{i\in A_{n,k}} \left(W_{n,k}^{(i)} - r_{n,k}\right), 0\right]. \tag{5.20}$$

The difference between the $Z_{n,k}$ and $Y_{n,k}$ queues is the new auxiliary variable $r_{n,k} \in [0, d_n]$ chosen at time t_k independent of the frame size T_k and per-frame class n arrivals $A_{n,k}$. Assume initially $Y_{n,0} = 0$ for all n. While the $Z_{n,k}$ queues help to attain delay constraints $\overline{W}_n \leq d_n$ for all classes n (as seen in Section 5.3), the $Y_{n,k}$ queues are useful to achieve the optimal delay vector $(\overline{W}_n^*)_{n=1}^N$.

Delay Fairness Policy DelayFair**:**

1. In every frame $k \in \mathbb{Z}^+$, prioritize job classes in the decreasing order of the ratio $(Z_{n,k} + Y_{n,k})/\mathbb{E}[S_n]$, where $\mathbb{E}[S_n]$ is the mean size of a class n job; ties are broken arbitrarily.

2. At the end of every frame $k \in \mathbb{Z}^+$, compute $Z_{n,k+1}$ and $Y_{n,k+1}$ for all classes n by (5.9) and (5.20), respectively, where $r_{n,k}$ is the solution to the one-variable convex program:

$$\text{minimize:} \quad V f_n(r_{n,k}) - Y_{n,k}\,\lambda_n\,r_{n,k} \tag{5.21}$$

$$\text{subject to:} \quad 0 \le r_{n,k} \le d_n \tag{5.22}$$

where $V > 0$ is a predefined control parameter.

The DelayFair policy needs the information of arrival rates λ_n and mean job sizes $\mathbb{E}[S_n]$ in all job classes; higher moments are not required. If penalty functions $f_n(\cdot)$ are differentiable, the closed-form solution to (5.21)-(5.22) is easily computed. In the example of delay proportional fairness with quadratic penalty functions $f_n(\overline{W}_n) = \frac{1}{2}\,c_n(\overline{W}_n)^2$ for all n, the second step of the DelayFair policy solves:

$$\begin{aligned} \text{minimize:} \quad & \frac{V\,c_n}{2}\,r_{n,k}^2 - Y_{n,k}\,\lambda_n\,r_{n,k} \\ \text{subject to:} \quad & 0 \le r_{n,k} \le d_n \end{aligned}$$

with the solution being $r_{n,k}^* = \min\left[d_n, \frac{Y_{n,k}\,\lambda_n}{V\,c_n}\right]$.

5.4.3 Construction of DelayFair

We first construct a Lyapunov drift inequality leading to the DelayFair policy. Intuitions are provided later. Define the Lyapunov function $L(\boldsymbol{Z}_k, \boldsymbol{Y}_k) \triangleq \frac{1}{2}\sum_{n=1}^{N}(Z_{n,k}^2 + Y_{n,k}^2)$ and one-frame Lyapunov drift $\Delta(\boldsymbol{Z}_k, \boldsymbol{Y}_k) \triangleq \mathbb{E}\left[L(\boldsymbol{Z}_{k+1}, \boldsymbol{Y}_{k+1}) - L(\boldsymbol{Z}_k, \boldsymbol{Y}_k) \mid \boldsymbol{Z}_k, \boldsymbol{Y}_k\right]$. Taking the square of (5.20) yields

$$Y_{n,k+1}^2 \leq \left[Y_{n,k} + \sum_{i \in A_{n,k}} \left(W_{n,k}^{(i)} - r_{n,k}\right)\right]^2. \tag{5.23}$$

Summing (5.10) and (5.23) over $n \in \{1, \ldots, N\}$, dividing the result by two, and taking conditional expectation on $\boldsymbol{Z}_k$ and $\boldsymbol{Y}_k$, we get

$$\begin{aligned} \Delta(\boldsymbol{Z}_k, \boldsymbol{Y}_k) \leq C - \sum_{n=1}^N Z_{n,k}\, d_n\, \mathbb{E}\left[|A_{n,k}| \mid \boldsymbol{Z}_k, \boldsymbol{Y}_k\right] - \sum_{n=1}^N Y_{n,k}\, \mathbb{E}\left[r_{n,k}\, |A_{n,k}| \mid \boldsymbol{Z}_k, \boldsymbol{Y}_k\right] \\ + \sum_{n=1}^N (Z_{n,k} + Y_{n,k})\, \mathbb{E}\left[\sum_{i \in A_{n,k}} W_{n,k}^{(i)} \mid \boldsymbol{Z}_k, \boldsymbol{Y}_k\right], \end{aligned} \tag{5.24}$$

where $C > 0$ is a finite constant, different from that used in Section 5.3.2, that upper bounds the sum of all $(\boldsymbol{Z}_k, \boldsymbol{Y}_k)$-independent terms in (5.24); the existence of C can be proved similarly as Lemma 5.19 of Section 5.10.2.

Adding to both sides of (5.24) the term $V \sum_{n=1}^N \mathbb{E}\left[f_n(r_{n,k})\, T_k \mid \boldsymbol{Z}_k, \boldsymbol{Y}_k\right]$, where $V > 0$ is a predefined control parameter, and evaluating the result under a scheduling policy defined in Assumption 5.1 (similar to the analysis in Section 5.3.2), we have the Lyapunov drift inequality:

$$\begin{aligned} &\Delta(\boldsymbol{Z}_k, \boldsymbol{Y}_k) + V \sum_{n=1}^N \mathbb{E}\left[f_n(r_{n,k})\, T_k \mid \boldsymbol{Z}_k, \boldsymbol{Y}_k\right] \\ &\leq \left(C - \mathbb{E}\left[T_k\right] \sum_{n=1}^N Z_{n,k}\, \lambda_n\, d_n\right) \\ &\quad + \mathbb{E}\left[T_k\right] \sum_{n=1}^N \mathbb{E}\left[V\, f_n(r_{n,k}) - Y_{n,k}\, \lambda_n\, r_{n,k} \mid \boldsymbol{Z}_k, \boldsymbol{Y}_k\right] \\ &\quad + \mathbb{E}\left[T_k\right] \sum_{n=1}^N (Z_{n,k} + Y_{n,k})\, \lambda_n\, \overline{W}_{n,k}, \end{aligned} \tag{5.25}$$

where $\overline{W}_{n,k}$ denotes the mean queueing delay of class n if the (random) control in frame k is repeated in every frame. Over all scheduling policies in Assumption 5.1 and all (possibly random) choices of $r_{n,k}$, we are interested in minimizing the right side of (5.25) in every frame. The result is the DelayFair policy. Recall that in this section we assume a constant service rate so that $\mathbb{E}[T_k]$ is fixed. The first and second step of the DelayFair policy minimizes the last term (by $c\mu$ rule) and the second-to-last term of (5.25), respectively.

5.4.4 Intuition on DelayFair

An intuition on minimizing an upper bound on the left side of (5.25), i.e., the DelayFair policy, is provided as follows. The first term $\Delta(\boldsymbol{Z}_k, \boldsymbol{Y}_k)$ of (5.25) is the expected growth of a queue size measure over virtual queues $Z_{n,k}$ and $Y_{n,k}$ over a frame. Similar to the last problem, minimizing $\Delta(\boldsymbol{Z}_k, \boldsymbol{Y}_k)$ in every frame shall stabilize all $Z_{n,k}$ and $Y_{n,k}$ queues, and all delay constraints $\overline{W}_n \le d_n$ are satisfied.

If all $Y_{n,k}$ queues are mean rate stable, similar to Lemma 5.7, we can show from (5.20) that $\overline{W}_n \le \overline{r}_n$ for all n, where

$$\overline{r}_n \triangleq \limsup_{K\to\infty} \frac{\mathbb{E}\left[\sum_{k=0}^{K-1} r_{n,k}\,|A_{n,k}|\right]}{\mathbb{E}\left[\sum_{k=0}^{K-1} |A_{n,k}|\right]},$$

defined similarly as $\overline{W}_n$ in (5.1). It follows that $\sum_{n=1}^N f_n(\overline{W}_n) \le \sum_{n=1}^N f_n(\overline{r}_n)$, because all $f_n(\cdot)$ functions are nondecreasing. The last inequality suggests that, provided all $Y_{n,k}$ queues are mean rate stable, we can minimize the objective function $\sum_{n=1}^N f_n(\overline{W}_n)$ by indirectly minimizing $\sum_{n=1}^N f_n(\overline{r}_n)$. The subtlety is that minimizing $\sum_{n=1}^N f_n(\overline{r}_n)$ is closely related to minimizing $\sum_{n=1}^N \mathbb{E}[f_n(r_{n,k})\,T_k \mid \boldsymbol{Z}_k, \boldsymbol{Y}_k]$ in every frame.

Minimizing both $\Delta(\boldsymbol{Z}_k, \boldsymbol{Y}_k)$ and $\sum_{n=1}^N \mathbb{E}[f_n(r_{n,k})\,T_k \mid \boldsymbol{Z}_k, \boldsymbol{Y}_k]$ in a frame, however, creates a tradeoff. Minimizing the former term needs large $r_{n,k}$ values because they are services of the $Y_{n,k}$ queues (see (5.20)). Yet, minimizing the later term needs small $r_{n,k}$ values since $f_n(\cdot)$ is nondecreasing. It is then natural to consider minimizing a weighted sum of them, which is the left side of (5.25), where the tradeoff is controlled by the V parameter. As we will see in the next section, having a large V value puts more weights on minimizing $\sum_{n=1}^{N} \mathbb{E}\left[f_n(r_{n,k})\, T_k \mid \boldsymbol{Z}_k, \boldsymbol{Y}_k\right]$, resulting in a convex delay penalty closer to optimal. The resulting tradeoff is that the $Z_{n,k}$ queues take longer to approach mean rate stability (see (5.28) later), resulting in a longer time to meeting the time average constraints $\overline{W}_n \leq d_n$.

5.4.5 Performance of DelayFair

Theorem 5.12. Given any feasible delay bounds $\{d_1, \ldots, d_N\}$ under scheduling policies defined in Assumption 5.1, the DelayFair policy satisfies all constraints $\overline{W}_n \leq d_n$ and yields convex delay cost satisfying

$$\limsup_{K\to\infty} \sum_{n=1}^{N} f_n\left(\frac{\mathbb{E}\left[\sum_{k=0}^{K-1}\sum_{i\in A_{n,k}} W_{n,k}^{(i)}\right]}{\mathbb{E}\left[\sum_{k=0}^{K-1} |A_{n,k}|\right]}\right) \leq \frac{C\sum_{n=1}^{N}\lambda_n}{V} + \sum_{n=1}^{N} f_n(\overline{W}_n^*),$$

where $V > 0$ is a predefined control parameter and $C > 0$ a finite constant. The convex delay cost can be made arbitrarily close to the optimal value $\sum_{n=1}^{N} f_n(\overline{W}_n^*)$ by choosing V sufficiently large.

Remark 5.13. The DelayFair policy can be viewed as a learning algorithm over the multi-class $M/G/1$ queue. It updates controls by observing past queueing delays in each job class, and requires limited statistical knowledge of the queue. The effectiveness of the learning algorithm is controlled by the V parameter: A large V yields performance (average delay penalty) closer to optimal, as seen in Theorem 5.12, at the expense of increasing the time to learn to meet the time-average constraints. Specifically, (5.28) shows that the convergence speed of the DelayFair policy is related to $\sqrt{2(C+VD)/K}$, where C and D are positive constants, and K is the number of passed busy periods. Similarly, our control policies for the two rate allocation problems presented later in Section 5.5 and 5.6 can also be viewed as learning algorithms that have a similar tradeoff between performance and learning time, controlled by the V parameter.

Proof (Theorem 5.12)**.** Consider the optimal stationary randomized priority policy π^*_{rand} that yields optimal average delays $\overline{W}^*_n \leq d_n$ for all classes n. Since the DelayFair policy minimizes the right side of (5.25) in every frame, if we compare DelayFair with policy π^*_{rand} and the genie decisions $r^*_{n,k} = \overline{W}^*_n$ for all n and k, (5.25) under DelayFair is further upper bounded by

$$\begin{aligned}
&\Delta(\boldsymbol{Z}_k, \boldsymbol{Y}_k) + V \sum_{n=1}^{N} \mathbb{E}\left[f_n(r_{n,k})\, T_k \mid \boldsymbol{Z}_k, \boldsymbol{Y}_k\right] \\
&\leq C - \mathbb{E}\left[T_k\right] \sum_{n=1}^{N} Z_{n,k}\, \lambda_n\, d_n + \mathbb{E}\left[T_k\right] \sum_{n=1}^{N} (Z_{n,k} + Y_{n,k}) \lambda_n \overline{W}^*_n \\
&\quad + \mathbb{E}\left[T_k\right] \sum_{n=1}^{N} \Big(V f_n(\overline{W}^*_n) - Y_{n,k}\, \lambda_n \overline{W}^*_n \Big) \\
&\leq C + V \mathbb{E}\left[T_k\right] \sum_{n=1}^{N} f_n(\overline{W}^*_n).
\end{aligned} \tag{5.26}$$

Removing the second term of (5.26) yields

$$\Delta(\boldsymbol{Z}_k, \boldsymbol{Y}_k) \leq C + V \mathbb{E}\left[T_k\right] \sum_{n=1}^{N} f_n(\overline{W}^*_n) \leq C + VD, \tag{5.27}$$

where $D \triangleq \mathbb{E}\left[T_k\right] \sum_{n=1}^{N} f_n(\overline{W}^*_n)$ is a finite constant. Taking expectation of (5.27), summing over $k \in \{0, \ldots, K-1\}$, and noting $L(\boldsymbol{Z}_0, \boldsymbol{Y}_0) = 0$, we get $\mathbb{E}\left[L(\boldsymbol{Z}_K, \boldsymbol{Y}_K)\right] \leq K(C + VD)$. It follows that, for every $Z_{n,k}$ queue, we have

$$0 \leq \frac{\mathbb{E}\left[Z_{n,K}\right]}{K} \leq \sqrt{\frac{\mathbb{E}\left[Z^2_{n,K}\right]}{K^2}} \leq \sqrt{\frac{2\mathbb{E}\left[L(\boldsymbol{Z}_k, \boldsymbol{Y}_K)\right]}{K^2}} \leq \sqrt{\frac{2C}{K} + \frac{2VD}{K}}, \quad \forall\, n \in \{1, \ldots, N\}. \tag{5.28}$$

Passing $K \to \infty$ proves mean rate stability for all $Z_{n,k}$ queues; therefore, all constraints $\overline{W}_n \leq d_n$ are satisfied by Lemma 5.7. Likewise, all $Y_{n,k}$ queues are mean rate stable.

Next, taking expectation of (5.26), summing over $k \in \{0, \ldots, K-1\}$, dividing by V, and noting $L(\boldsymbol{Z}_0, \boldsymbol{Y}_0) = 0$, we get

$$\frac{\mathbb{E}\left[L(\boldsymbol{Z}_K, \boldsymbol{Y}_K)\right]}{V} + \sum_{n=1}^{N} \mathbb{E}\left[\sum_{k=0}^{K-1} f_n(r_{n,k})\, T_k\right] \leq \frac{KC}{V} + \mathbb{E}\left[\sum_{k=0}^{K-1} T_k\right] \sum_{n=1}^{N} f_n(\overline{W}_n^*).$$

Removing the first term and dividing the rest by $\mathbb{E}\left[\sum_{k=0}^{K-1} T_k\right]$ yields

$$\sum_{n=1}^{N} \frac{\mathbb{E}\left[\sum_{k=0}^{K-1} f_n(r_{n,k})\, T_k\right]}{\mathbb{E}\left[\sum_{k=0}^{K-1} T_k\right]} \leq \frac{KC}{V\mathbb{E}\left[\sum_{k=0}^{K-1} T_k\right]} + \sum_{n=1}^{N} f_n(\overline{W}_n^*) \overset{(a)}{\leq} \frac{C\sum_{n=1}^{N} \lambda_n}{V} + \sum_{n=1}^{N} f_n(\overline{W}_n^*), \tag{5.29}$$

where (a) follows $\mathbb{E}\left[T_k\right] \geq \mathbb{E}\left[I_k\right] = 1/(\sum_{n=1}^{N} \lambda_n)$ for all k. By a generalized Jensen's inequality [Neely 2010c, Lemma 7.6] and convexity of $f_n(\cdot)$, we get

$$\sum_{n=1}^{N} \frac{\mathbb{E}\left[\sum_{k=0}^{K-1} f_n(r_{n,k})\, T_k\right]}{\mathbb{E}\left[\sum_{k=0}^{K-1} T_k\right]} \geq \sum_{n=1}^{N} f_n\left(\frac{\mathbb{E}\left[\sum_{k=0}^{K-1} r_{n,k}\, T_k\right]}{\mathbb{E}\left[\sum_{k=0}^{K-1} T_k\right]}\right). \tag{5.30}$$

Combining (5.29) and (5.30), and taking a lim sup as $K \to \infty$, we get

$$\limsup_{K\to\infty} \sum_{n=1}^{N} f_n\left(\frac{\mathbb{E}\left[\sum_{k=0}^{K-1} r_{n,k}\, T_k\right]}{\mathbb{E}\left[\sum_{k=0}^{K-1} T_k\right]}\right) \leq \frac{C\sum_{n=1}^{N} \lambda_n}{V} + \sum_{n=1}^{N} f_n(\overline{W}_n^*).$$

The next lemma completes the proof. □

Lemma 5.14 (Proof in Section 5.10.3)**.** If all $Y_{n,k}$ queues are mean rate stable, then

$$\limsup_{K\to\infty} \sum_{n=1}^{N} f_n\left(\frac{\mathbb{E}\left[\sum_{k=0}^{K-1} \sum_{i\in A_{n,k}} W_{n,k}^{(i)}\right]}{\mathbb{E}\left[\sum_{k=0}^{K-1} |A_{n,k}|\right]}\right) \leq \limsup_{K\to\infty} \sum_{n=1}^{N} f_n\left(\frac{\mathbb{E}\left[\sum_{k=0}^{K-1} r_{n,k}\, T_k\right]}{\mathbb{E}\left[\sum_{k=0}^{K-1} T_k\right]}\right).$$

5.5 Delay-Constrained Optimal Rate Control

In this section we incorporate dynamic allocations of the service rate into the queueing system. As specified in Assumption 5.2, we focus on frame-based policies that allocate a fixed service rate $\mu(P_k)$ with a per-unit-time service cost $P_k \in [P_{\min}, P_{\max}]$ in the kth busy period. No service rates are allocated when the system is idle. Here, the frame size T_k, busy period B_k, per-frame class n arrivals $A_{n,k}$, and queueing delays $W_{n,k}^{(i)}$, are all functions of P_k. Similar to the definition of average queueing delay in (5.1), we define the average service cost

$$\overline{P} \triangleq \limsup_{K\to\infty} \frac{\mathbb{E}\left[\sum_{k=0}^{K-1} P_k\, B_k(P_k)\right]}{\mathbb{E}\left[\sum_{k=0}^{K-1} T_k(P_k)\right]}, \tag{5.31}$$

where $B_k(P_k)$ and $T_k(P_k)$ emphasize the dependence of B_k and T_k on P_k. It is easy to show that $B_k(P_k)$ and $T_k(P_k)$ are decreasing in P_k. The goal is to solve the delay-constrained optimal rate control problem:

$$\text{minimize:} \quad \overline{P} \tag{5.32}$$

$$\text{subject to:} \quad \overline{W}_n \le d_n, \quad \forall\, n \in \{1, \dots, N\} \tag{5.33}$$

over scheduling and rate allocation policies defined in Assumption 5.1 and 5.2.

From Lemma 5.4 and Assumption 5.2, every feasible performance vector of average service cost and mean queue delay in all job classes is achieved by a stationary randomized policy that randomly chooses a strict priority policy and a fixed rate allocation in every busy period according to a stationary distribution. Therefore, without loss of generality, it suffices to solve (5.32)-(5.33) over the class of frame-based policies that choose a strict priority policy and a fixed rate allocation in every busy period; the decisions are possibly random. A stationary randomized policy is a special case in this class of policies. Let π^* denote the optimal stationary randomized policy that yields optimal average service cost P^* and feasible mean queueing delay $\overline{W}_n^* \le d_n$ in every class $n \in \{1, \dots, N\}$.

5.5.1 Dynamic Rate Control Policy DynRate

We present the policy that solves (5.32)-(5.33). We setup the same virtual queues $Z_{n,k}$ as in (5.9) to tackle delay constraints $\overline{W}_n \leq d_n$; we assume initially $Z_{n,0} = 0$ for all n.

Dynamic Rate Control (DynRate) Policy:

1. In frame $k \in \mathbb{Z}^+$, use the strict priority policy, denoted by π_k, that prioritizes job classes in the decreasing order of $Z_{n,k}/\mathbb{E}[S_n]$, where $\mathbb{E}[S_n]$ is the mean job size of class n; ties are broken arbitrarily.
2. In the busy period of frame $k \in \mathbb{Z}^+$, allocate service rate $\mu(P_k)$ that minimizes the weighted sum of service cost and rate-dependent average queueing delays:

$$\text{minimize:} \quad \left(V \sum_{n=1}^{N} \lambda_n \mathbb{E}[S_n]\right) \frac{P_k}{\mu(P_k)} + \sum_{n=1}^{N} Z_{n,k}\, \lambda_n \overline{W}_n(\pi_k, P_k) \tag{5.34}$$

$$\text{subject to:} \quad P_k \in [P_{\min}, P_{\max}] \tag{5.35}$$

   where $\overline{W}_n(\pi_k, P_k)$ is the mean queueing delay of class n when service rate $\mu(P_k)$ and strict priority policy π_k are used in all busy periods. The value of $\overline{W}_n(\pi_k, P_k)$ is given in (5.8) after proper re-ordering of job classes according to policy π_k.

3. Update all $Z_{n,k}$ queues by (5.9) at every frame boundary.

The DynRate policy requires the knowledge of arrival rates and the first two moments of job sizes. We can remove its dependence on the second moments of job sizes so that the policy depends only on first-order statistics; see Section 5.10.4 for details.

5.5.2 Intuition on DynRate

We construct a Lyapunov drift inequality first, and provide intuitions later. Define the Lyapunov function $L(\boldsymbol{Z}_k) = \frac{1}{2}\sum_{n=1}^{N} Z_{n,k}^2$ and one-frame Lyapunov drift $\Delta(\boldsymbol{Z}_k) = \mathbb{E}\left[L(\boldsymbol{Z}_{k+1}) - L(\boldsymbol{Z}_k) \mid \boldsymbol{Z}_k\right]$. Following the analysis in Section 5.3.2, we have

$$\Delta(\boldsymbol{Z}_k) \leq C + \sum_{n=1}^{N} Z_{n,k}\,\mathbb{E}\left[\sum_{i\in A_{n,k}} \left(W_{n,k}^{(i)} - d_n\right) \mid \boldsymbol{Z}_k\right]. \tag{5.36}$$

Adding $V\mathbb{E}\left[P_k\, B_k(P_k) \mid \boldsymbol{Z}_k\right]$ to both sides of (5.36), where $V > 0$ is a control parameter, we get

$$\Delta(\boldsymbol{Z}_k) + V\mathbb{E}\left[P_k\, B_k(P_k) \mid \boldsymbol{Z}_k\right] \leq C + \Phi(\boldsymbol{Z}_k), \tag{5.37}$$

where

$$\Phi(\boldsymbol{Z}_k) \triangleq \mathbb{E}\left[V P_k\, B_k(P_k) + \sum_{n=1}^{N} Z_{n,k} \sum_{i\in A_{n,k}} (W_{n,k}^{(i)} - d_n) \mid \boldsymbol{Z}_k\right].$$

We are interested in the scheduling and rate control policy that minimizes the ratio

$$\frac{\Phi(\boldsymbol{Z}_k)}{\mathbb{E}\left[T_k(P_k) \mid \boldsymbol{Z}_k\right]}. \tag{5.38}$$

over frame-based policies in every frame k. In the denominator of (5.38), the frame size $T_k(P_k)$ depends on $\boldsymbol{Z}_k$ because the assignment of service cost P_k that affects $T_k(P_k)$ may be $\boldsymbol{Z}_k$-dependent; for a given P_k, $T_k(P_k)$ is independent of $\boldsymbol{Z}_k$.

The intuition on minimizing (5.38) is as follows. Similar to previous problems, minimizing the Lyapunov drift $\Delta(\boldsymbol{Z}_k)$ in every frame helps to achieve delay constraints $\overline{W}_n \leq d_n$ in all classes. An added component here is that we can increase service rate (and service cost) to improve queueing delay which decreases $\Delta(\boldsymbol{Z}_k)$. Thus, there is a tradeoff between service cost and stability of the $Z_{n,k}$ queues, captured by the left side of (5.37). If we follow what we have done in previous problems, we shall minimize the right size of (5.37), i.e., $\Phi(\boldsymbol{Z}_k)$, in every frame. That is not enough here, however, because the frame size depends on the chosen service cost. The modification is to instead minimize the ratio of $\Phi(\boldsymbol{Z}_k)$ over the mean frame size, namely (5.38).

5.5.3 Construction of DynRate

We simplify (5.38) to show that the DynRate policy is the desired policy. Lemma 5.15, given next, shows that the minimizer of (5.38) is a deterministic rate allocation and strict priority policy. Specifically, we may consider every $p \in \mathcal{P}$ in Lemma 5.15 as one such deterministic policy, and the random variable P is a frame-based policy making possibly random control decisions in every frame.

Lemma 5.15. Let P be a continuous random variable with state space $\mathcal{P}$. Let G and H be two random variables that depend on the realization of P such that, for each $p \in \mathcal{P}$, $G(p)$ and $H(p)$ are well-defined random variables. Define

$$p^* \triangleq \operatorname{argmin}_{p\in\mathcal{P}} \frac{\mathbb{E}\left[G(p)\right]}{\mathbb{E}\left[H(p)\right]}, \quad U^* \triangleq \frac{\mathbb{E}\left[G(p^*)\right]}{\mathbb{E}\left[H(p^*)\right]}.$$

Then $\frac{\mathbb{E}[G]}{\mathbb{E}[H]} \geq U^*$ regardless of the distribution of P.

Proof (Lemma 5.15)**.** For each $p \in \mathcal{P}$, we have $\frac{\mathbb{E}[G(p)]}{\mathbb{E}[H(p)]} \geq U^*$. Then

$$\frac{\mathbb{E}\left[G\right]}{\mathbb{E}\left[H\right]} = \frac{\mathbb{E}_P\left[\mathbb{E}\left[G(p)\right]\right]}{\mathbb{E}_P\left[\mathbb{E}\left[H(p)\right]\right]} \geq \frac{\mathbb{E}_P\left[U^*\mathbb{E}\left[H(p)\right]\right]}{\mathbb{E}_P\left[\mathbb{E}\left[H(p)\right]\right]} = U^*,$$

which is independent of the distribution of P. $\square$

Under a fixed service rate $\mu(P_k)$ and a strict priority rule, (5.38) is equal to

$$\begin{aligned}\frac{\Phi(\boldsymbol{Z}_k)}{\mathbb{E}\left[T_k(P_k) \mid \boldsymbol{Z}_k\right]} &= \frac{V\mathbb{E}\left[P_k B_k(P_k)\right] + \sum_{n=1}^{N} Z_{n,k}\,\lambda_n(\overline{W}_{n,k}(P_k) - d_n)\mathbb{E}\left[T_k(P_k)\right]}{\mathbb{E}\left[T_k(P_k)\right]} \\ &= V P_k \frac{\sum_{n=1}^{N} \lambda_n \mathbb{E}\left[S_n\right]}{\mu(P_k)} + \sum_{n=1}^{N} Z_{n,k}\,\lambda_n(\overline{W}_{n,k}(P_k) - d_n), \qquad (5.39)\end{aligned}$$

where we have used, by renewal theory,

$$\frac{\mathbb{E}\left[B_k(P_k)\right]}{\mathbb{E}\left[T_k(P_k)\right]} = \sum_{n=1}^{N} \rho_n(P_k) = \sum_{n=1}^{N} \lambda_n \frac{\mathbb{E}\left[S_n\right]}{\mu(P_k)}.$$

It follows that our desired policy minimizes

$$\left(V \sum_{n=1}^{N} \lambda_n \mathbb{E}\left[S_n\right]\right) \frac{P_k}{\mu(P_k)} + \sum_{n=1}^{N} Z_{n,k}\, \lambda_n \overline{W}_{n,k}(P_k) \tag{5.40}$$

in every frame k over $P_k \in [P_{\min}, P_{\max}]$ and strict priority policies.

To further simplify, under a fixed service rate $\mu(P_k)$, the second term of (5.40) is minimized by assigning priorities in the decreasing order of $Z_{n,k}/\mathbb{E}\left[S_n\right]$ according to the $c\mu$ rule (noting that minimizing a linear function over strict priority policies is equivalent to minimizing over all stationary randomized priority policies, because a vertex of the delay polymatroid attains the minimum). This strict priority policy, denoted by π_k, is optimal regardless of the value of P_k, and thus is overall optimal. Interestingly, priority assignment is decoupled from optimal rate allocation. Under policy π_k, solving (5.34)-(5.35) reveals the optimal rate allocation in frame k. These discussions lead to the DynRate policy.

5.5.4 Performance of DynRate

Theorem 5.16. Let P^* be the optimal average service cost for the problem (5.32)-(5.33). The DynRate policy satisfies all delay constraints $\overline{W}_n \leq d_n$ and attains average service cost $\overline{P}$ satisfying

$$\overline{P} \leq \frac{C \sum_{n=1}^{N} \lambda_n}{V} + P^*,$$

where $C > 0$ is a finite constant and $V > 0$ a predefined control parameter. The gap between $\overline{P}$ and the optimal P^* can be made arbitrarily small by a sufficiently large V.

Proof (Theorem 5.16)**.** Consider the optimal stationary randomized policy π^* that yields optimal average cost P^* and feasible mean queueing delay $\overline{W}_n^* \leq d_n$ in all classes. Let

P_k^* denote its service cost allocation in frame k. Since policy π^* makes i.i.d. decisions over frames, by renewal reward theory we have

$$P^* = \frac{\mathbb{E}\left[P_k^* B(P_k^*)\right]}{\mathbb{E}\left[T(P_k^*)\right]}.$$

The ratio (5.38) under policy π^* is equal to (cf. (5.39))

$$V\frac{\mathbb{E}\left[P_k^* B(P_k^*)\right]}{\mathbb{E}\left[T(P_k^*)\right]} + \sum_{n=1}^{N} Z_{n,k}\, \lambda_n \left(\overline{W}_n^* - d_n\right) \leq V P^*.$$

Since the DynRate policy minimizes (5.38) over frame-based policies that update controls over busy periods, including the optimal π^*, (5.38) under the DynRate policy satisfies

$$\frac{\Phi(\boldsymbol{Z}_k)}{\mathbb{E}\left[T_k(P_k) \mid \boldsymbol{Z}_k\right]} \leq V P^*,\ \forall\, k \in \mathbb{Z}^+.$$

Using this bound, (5.37) under the DynRate policy satisfies

$$\Delta(\boldsymbol{Z}_k) + V\mathbb{E}\left[P_k\, B_k(P_k) \mid \boldsymbol{Z}_k\right] \leq C + V P^*\, \mathbb{E}\left[T_k(P_k) \mid \boldsymbol{Z}_k\right].$$

Taking expectation, summing over $k \in \{0, \dots, K-1\}$, and noting $L(\boldsymbol{Z}_0) = 0$ yields

$$\mathbb{E}\left[L(\boldsymbol{Z}_K)\right] + V\sum_{k=0}^{K-1} \mathbb{E}\left[P_k\, B_k(P_k)\right] \leq KC + VP^*\, \mathbb{E}\left[\sum_{k=0}^{K-1} T_k(P_k)\right]. \tag{5.41}$$

Since $\mathbb{E}\left[T_k(P_k)\right]$ is decreasing in P_k and independent of scheduling policies under a fixed rate allocation, we get $\mathbb{E}\left[T_k(P_k)\right] \leq \mathbb{E}\left[T_0(P_{\min})\right]$ for all k. It follows that

$$\mathbb{E}\left[L(\boldsymbol{Z}_K)\right] + V\sum_{k=0}^{K-1} \mathbb{E}\left[P_k\, B_k(P_k)\right] \leq K(C + VP^*\, \mathbb{E}\left[T_0(P_{\min})\right]).$$

Removing the second term and dividing by K^2 yields

$$\frac{\mathbb{E}\left[L(\boldsymbol{Z}_K)\right]}{K^2} \leq \frac{C + VP^*\, \mathbb{E}\left[T_0(P_{\min})\right]}{K}.$$

Combining it with

$$0 \le \frac{\mathbb{E}[Z_{n,K}]}{K} \le \sqrt{\frac{\mathbb{E}\left[Z_{n,K}^2\right]}{K^2}} \le \sqrt{\frac{2\mathbb{E}[L(\boldsymbol{Z}_K)]}{K^2}}, \quad \forall n \in \{1, \dots, N\}$$

and passing $K \to \infty$ prove that all $Z_{n,k}$ queues are mean rate stable; all delay constraints $\overline{W}_n \le d_n$ are satisfied (by Lemma 5.7).

Next, removing the first term in (5.41) and dividing the result by $V\mathbb{E}\left[\sum_{k=0}^{K-1} T_k(P_k)\right]$ yields

$$\frac{\mathbb{E}\left[\sum_{k=0}^{K-1} P_k\, B_k(P_k)\right]}{\mathbb{E}\left[\sum_{k=0}^{K-1} T_k(P_k)\right]} \le \frac{C}{V} \frac{K}{\mathbb{E}\left[\sum_{k=0}^{K-1} T_k(P_k)\right]} + P^* \overset{(a)}{\le} \frac{C\sum_{n=1}^{N} \lambda_n}{V} + P^*,$$

where (a) uses $\mathbb{E}[T_k(P_k)] \ge \mathbb{E}[I_k] = 1/(\sum_{n=1}^{N} \lambda_n)$. Passing $K \to \infty$ finishes the proof. $\square$

5.6 Cost-Constrained Convex Delay Optimization

In the fourth problem, we minimize a separable convex function of the mean queueing delay vector subject to an average service cost constraint:

$$\text{minimize:} \quad \sum_{n=1}^{N} f_n(\overline{W}_n) \tag{5.42}$$

$$\text{subject to:} \quad \overline{P} \le P_{\text{const}} \tag{5.43}$$

over scheduling and rate allocation policies defined in Assumption 5.1 and 5.2. Without loss of generality, we shall focus on solving (5.42)-(5.43) over frame-based policies. This is similar to the previous rate control problem. The term $\overline{P}$ is defined in (5.31), and $P_{\text{const}} > 0$ is a given feasible bound. We assume all $f_n(\cdot)$ functions are nondecreasing, nonnegative, continuous, and convex.

Here, the set of feasible performance vectors of average service cost and average queueing delay in all job classes is complicated because feasible mean delays are indirectly decided by the service cost constraint (5.43). Yet, knowing that there exists an optimal stationary randomized policy, similar to the previous rate control problem, we can construct a frame-based policy that solves (5.42)-(5.43).

We setup the same virtual queues $\{Y_{n,k}\}_{k=0}^{\infty}$ for all classes $n \in \{1, \ldots, N\}$ as in (5.20), except that the auxiliary variables $r_{n,k}$ take values in a new region $[0, R_{\max,n}]$, for some $R_{\max,n} > 0$ sufficiently large. Specifically, we need $R_{\max,n}$ to be greater than the optimal delay $\overline{W}_n^*$ in (5.42)-(5.43) in each class n; we may define $R_{\max,n}$ as the maximum average delay over all classes under the minimum rate allocation $\mu(P_{\min})$.

We define a *virtual cost queue* $\{X_k\}_{k=0}^{\infty}$ that evolves at frame boundaries $\{t_k\}_{k=0}^{\infty}$ as

$$X_{k+1} = \max\left[X_k + P_k B_k(P_k) - P_{\text{const}} T_k(P_k),\, 0\right]. \tag{5.44}$$

Assume $X_0 = 0$. The virtual cost queue X_k helps to achieve the average cost constraint $\overline{P} \leq P_{\text{const}}$.

Lemma 5.17 (Proof in Section 5.10.5)**.** If the virtual cost queue X_k is mean rate stable, then $\overline{P} \leq P_{\text{const}}$.

5.6.1 Cost-Constrained Delay Fairness Policy CostDelayFair

The next policy solves (5.42)-(5.43); it has similar steps as those in the DelayFair and the DynRate policy.

Cost-Constrained Delay Fairness (CostDelayFair) **Policy:**

1. In every frame $k \in \mathbb{Z}^+$, observe X_k and $Y_{n,k}$ for all n at the beginning the frame, and use the strict priority policy π_k that assigns priorities in

the decreasing order of $Y_{n,k}/\mathbb{E}[S_n]$; ties are broken arbitrarily. Further, in the busy period of frame k, allocate service rate $\mu(P_k)$ that solves:

$$\begin{aligned} \text{minimize:} \quad & \frac{P_k}{\mu(P_k)} \left[X_k \sum_{n=1}^{N} \lambda_n \mathbb{E}[S_n] \right] + \sum_{n=1}^{N} Y_{n,k} \, \lambda_n \overline{W}_n(\pi_k, P_k) \\ \text{subject to:} \quad & P_k \in [P_{\min}, P_{\max}], \end{aligned}$$

where $\overline{W}_n(\pi_k, P_k)$ is the mean queueing delay of class n when policy π_k and service rate $\mu(P_k)$ are used in all busy periods. The term $\overline{W}_n(\pi_k, P_k)$ is defined in (5.8) after proper re-ordering of job classes according to π_k.

2. At frame boundaries, update $Y_{n,k}$ for all classes n and X_k by (5.20) and (5.44), respectively. In (5.20), the auxiliary variable $r_{n,k}$ is the solution to the one-variable convex program

$$\begin{aligned} \text{minimize:} \quad & V f_n(r_{n,k}) - Y_{n,k} \, \lambda_n \, r_{n,k} \\ \text{subject to:} \quad & 0 \le r_{n,k} \le R_{\max,n} \end{aligned}$$

which is easily solved if $f_n(\cdot)$ are differentiable.

5.6.2 Construction of CostDelayFair

The construction of the CostDelayFair policy follows closely with those in previous problems; details are omitted for brevity. Define the vector $\chi_k = [X_k; Y_{1,k}, \ldots, Y_{N,k}]$, the Lyapunov function $L(\chi_k) \triangleq \frac{1}{2}(X_k^2 + \sum_{n=1}^{N} Y_{n,k}^2)$, and the one-frame Lyapunov drift $\Delta(\chi_k) \triangleq \mathbb{E}[L(\chi_{k+1}) - L(\chi_k) \mid \chi_k]$. There exists a finite constant $C > 0$ such that

$$\begin{aligned} \Delta(\chi_k) \le C &+ X_k \mathbb{E}[P_k B_k(P_k) - P_{\text{const}} \, T_k(P_k) \mid \chi_k] \\ &+ \sum_{n=1}^{N} Y_{n,k} \, \mathbb{E}\left[\sum_{i \in A_{n,k}} \left(W_{n,k}^{(i)} - r_{n,k} \right) \mid \chi_k \right]. \end{aligned} \tag{5.45}$$

Adding $V \sum_{n=1}^{N} \mathbb{E}\left[f_n(r_{n,k})\, T_k(P_k) \mid \chi_k\right]$ to both sides of (5.45), where $V > 0$ is a control parameter, and evaluating the result under a frame-based policy yields

$$\Delta(\chi_k) + V \sum_{n=1}^{N} \mathbb{E}\left[f_n(r_{n,k})\, T_k(P_k) \mid \chi_k\right] \leq C + \Psi(\chi_k), \tag{5.46}$$

where

$$\begin{aligned}\Psi(\chi_k) \triangleq\ & \mathbb{E}\left[T_k(P_k) \mid \chi_k\right] \sum_{n=1}^{N} Y_{n,k}\, \lambda_n \overline{W}_{n,k}(P_k) \\ & + X_k \mathbb{E}\left[P_k B_k(P_k) \mid \chi_k\right] - X_k P_{\text{const}}\, \mathbb{E}\left[T_k(P_k) \mid \chi_k\right] \\ & + \mathbb{E}\left[T_k(P_k) \mid \chi_k\right] \sum_{n=1}^{N} \mathbb{E}\left[V f_n(r_{n,k}) - Y_{n,k}\, \lambda_n\, r_{n,k} \mid \chi_k\right],\end{aligned}$$

where $\overline{W}_{n,k}(P_k)$ is the mean queueing delay of class n when the control in frame k is repeated in every frame.

We consider the policy that minimizes the ratio

$$\frac{\Psi(\chi_k)}{\mathbb{E}\left[T_k(P_k) \mid \chi_k\right]} \tag{5.47}$$

over frame-based policies in every frame k. Lemma 5.15 shows that the minimizer is a strict priority policy with a fixed rate allocation, under which the ratio (5.47) is equal to

$$\sum_{n=1}^{N} Y_{n,k}\, \lambda_n \overline{W}_{n,k}(P_k) + X_k \left(P_k\, \rho_{\text{sum}}(P_k) - P_{\text{const}}\right) + \sum_{n=1}^{N} \left(V f_n(r_{n,k}) - Y_{n,k}\, \lambda_n\, r_{n,k}\right),$$

where $\rho_{\text{sum}}(P_k) \triangleq \sum_{n=1}^{N} \lambda_n \mathbb{E}\left[S_n\right] / \mu(P_k)$. Under similar simplifications as those for the DynRate policy in Section 5.5, the CostDelayFair policy is the desired policy.

5.6.3 Performance of CostDelayFair

Theorem 5.18 (Proof in Section 5.10.6)**.** For any feasible average service cost constraint $\overline{P} \leq P_{\text{const}}$ in (5.42)-(5.43), let $(\overline{W}_n^*)_{n=1}^N$ be the optimal mean queueing delay vector. The CostDelayFair policy satisfies $\overline{P} \leq P_{\text{const}}$ and yields average delay penalty satisfying

$$\limsup_{K\to\infty} \sum_{n=1}^{N} f_n \left(\frac{\mathbb{E}\left[\sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)}\right]}{\mathbb{E}\left[\sum_{k=0}^{K-1} |A_{n,k}|\right]} \right) \leq \frac{C \sum_{n=1}^{N} \lambda_n}{V} + \sum_{n=1}^{N} f_n(\overline{W}_n^*), \tag{5.48}$$

where $V > 0$ is a predefined control parameter.

5.7 Simulations

We simulate all four control policies in a two-class nonpreemptive $M/G/1$ queue. We denote by $\mathcal{W}(P)$ the performance region of mean queueing delay under a constant service rate $\mu(P)$. Define $\rho_n \triangleq \lambda_n \mathbb{E}[X_n]$ and $R \triangleq \frac{1}{2}\sum_{n=1}^{2} \lambda_n \mathbb{E}\left[X_n^2\right]$, where $X_n = S_n/\mu(P)$ is the random service time of a class n job. We have [Bertsekas and Gallager 1992]

$$\mathcal{W}(P) = \left\{ (\overline{W}_1, \overline{W}_2) \;\middle|\; \begin{array}{l} \overline{W}_1 \geq \dfrac{R}{1-\rho_1},\ \overline{W}_2 \geq \dfrac{R}{1-\rho_2}, \\ \rho_1 \overline{W}_1 + \rho_2 \overline{W}_2 = \dfrac{(\rho_1+\rho_2)R}{1-\rho_1-\rho_2} \end{array} \right\}. \tag{5.49}$$

In (5.49), the two inequalities capture that the mean queueing delay in one class is minimized when it has priority over the other class. The equality is the $M/G/1$ conservation law [Kleinrock 1976].

Every simulation result in this section is a sample average over 10 runs, each of which lasts for 10^6 frames.

5.7.1 DelayFeas and DelayFair Policy

To simulate the first two DelayFeas and DelayFair policy, we consider a two-class $M/M/1$ queue with arrival rates $(\lambda_1, \lambda_2) = (1, 2)$ and mean service times $(\mathbb{E}[X_1], \mathbb{E}[X_2]) =$

$(0.4, 0.2)$; we consider service time directly instead of job sizes because there is no rate control. The performance region of mean queueing delay, using (5.49), is

$$\mathcal{W} = \left\{ (\overline{W}_1, \overline{W}_2) \;\middle|\; \overline{W}_1 + \overline{W}_2 = 2.4,\ \overline{W}_1 \geq 0.4,\ \overline{W}_2 \geq 0.4 \right\}. \tag{5.50}$$

DelayFeas **Policy**

In DelayFeas policy, we consider five sets of delay constraints $(d_1, d_2) = (0.45, 2.05)$, $(0.85, 1.65)$, $(1.25, 1.25)$, $(1.65, 0.85)$, and $(2.05, 0.45)$; they are all $(0.05, 0.05)$ entrywise larger than a feasible point on $\mathcal{W}$. The simulation results in Fig. 5.1 show that the DelayFeas policy adaptively yields feasible average delays in response to different constraints. Over the 10 simulation runs in each of the five cases, the sample standard deviation of the average delay in each job class is at most 0.017. Therefore, different simulation runs produce consistent results.

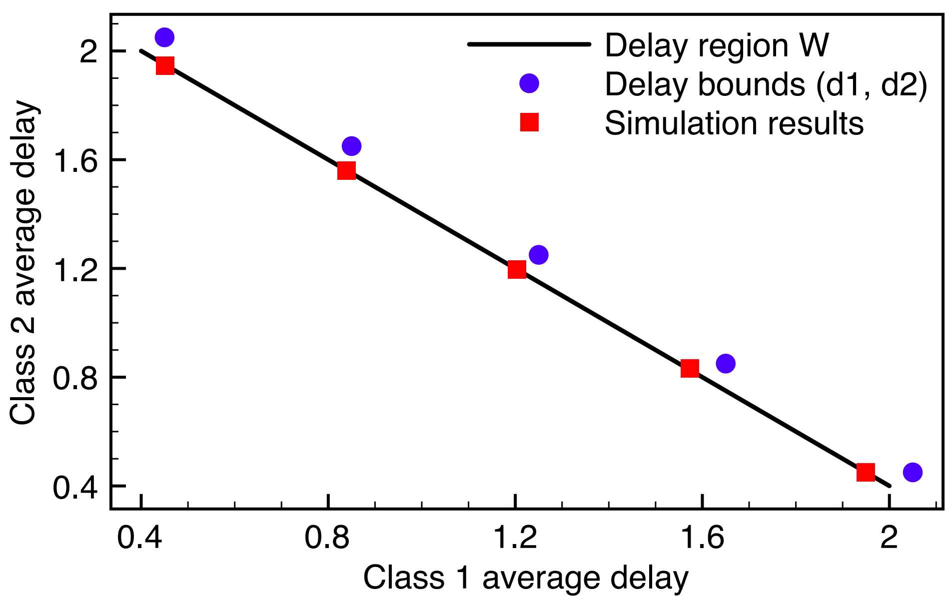

Figure 5.1: Simulation for the DelayFeas policy under different delay constraints (d_1, d_2).

DelayFair **Policy**

For the DelayFair policy, we consider the delay proportional fairness problem:

$$\text{minimize:} \quad \frac{1}{2}\,(\overline{W}_1)^2 + 2\,(\overline{W}_2)^2 \tag{5.51}$$

$$\text{subject to:} \quad (\overline{W}_1, \overline{W}_2) \in \mathcal{W} \tag{5.52}$$

$$\overline{W}_1 \leq 1.95, \overline{W}_2 \leq 1 \tag{5.53}$$

where $\mathcal{W}$ is given in (5.50). The optimal solution to (5.51)-(5.53) is $(\overline{W}_1^*, \overline{W}_2^*) = (1.92, 0.48)$, and the optimal delay penalty is $\frac{1}{2}(\overline{W}_1^*)^2 + 2(\overline{W}_2^*)^2 = 2.304$. We simulate the DelayFair policy for different values of control parameter V, and the results are in Table 5.1. The values in parentheses in Table 5.1 are sample standard deviations over the 10 simulation runs. As V increases, the DelayFair policy yields mean delay penalty approaching optimal.

V	$\overline{W}_1$	$\overline{W}_2$	$\frac{1}{2}\,(\overline{W}_1)^2 + 2\,(\overline{W}_2)^2$
100	1.6607 (0.0055)	0.7424 (0.0052)	2.4814 (0.0239)
1000	1.7977 (0.0057)	0.5984 (0.0043)	2.3321 (0.0199)
2000	1.8339 (0.0056)	0.5639 (0.0053)	2.3176 (0.0217)
5000	1.8679 (0.0073)	0.5276 (0.0050)	2.3014 (0.0222)
Optimal:	1.92	0.48	2.304

Table 5.1: Simulation for the DelayFair policy under different values of V.

5.7.2 DynRate **and** CostDelayFair **Policy**

In the two rate control problems, we consider a two-class $M/G/1$ queue with arrival rates $(\lambda_1, \lambda_2) = (1, 2)$. The size of a class 1 job is 0.5 with probability 0.8 and 3 otherwise. The size of a class 2 job is always one. The feasible expenditure of service cost in every busy period is in the discrete set $P \in \{16, 25\}$. We consider the service rate $\mu(P) = \sqrt{P}$.

Using (5.49), the full performance region of mean queueing delay, denoted by $\mathcal{W}$, is the convex hull of the two individual regions (see Fig. 5.2):

$$\mathcal{W}(16) = \left\{ (\overline{W}_1, \overline{W}_2) \,\middle|\, \overline{W}_1 + 2\overline{W}_2 = \frac{3}{2},\ \overline{W}_1 \geq \frac{1}{6},\ \overline{W}_2 \geq \frac{1}{4} \right\},$$
$$\mathcal{W}(25) = \left\{ (\overline{W}_1, \overline{W}_2) \,\middle|\, \overline{W}_1 + 2\overline{W}_2 = \frac{3}{5},\ \overline{W}_1 \geq \frac{1}{10},\ \overline{W}_2 \geq \frac{2}{15} \right\},$$

where $\mathcal{W}(16)$ and $\mathcal{W}(25)$ are the mean queueing delay regions under a constant service cost of $P = 16$ and $P = 25$, respectively.

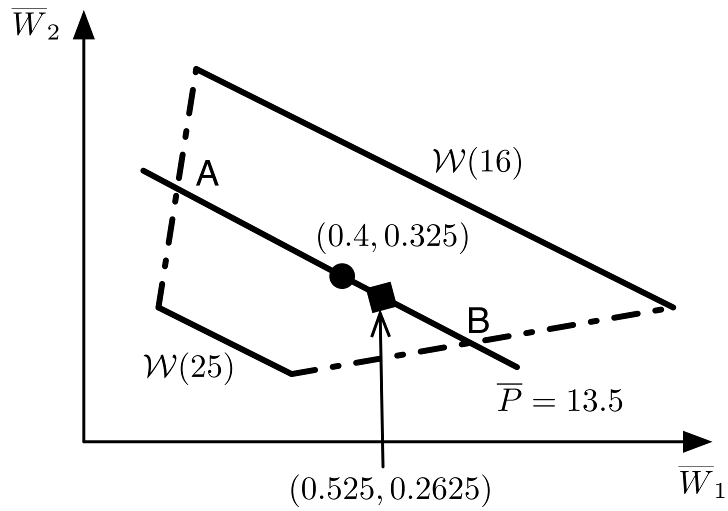

Figure 5.2: The performance region $\mathcal{W}$ of mean queueing delay in the simulations for the DynRate and the CostDelayFair policy.

DynRate **Policy**

For the DynRate policy, we solve

$$\text{minimize:} \quad \overline{P} \tag{5.54}$$
$$\text{subject to:} \quad (\overline{W}_1, \overline{W}_2) \in \mathcal{W} \tag{5.55}$$
$$\overline{W}_1 \leq 0.4,\ \overline{W}_2 \leq 0.325 \tag{5.56}$$

where $\mathcal{W}$ is the full queueing delay region in Fig. 5.2. The minimum average service cost is achieved by satisfying the constraints (5.56) with equality. By finding the stationary randomized policy that yields $(\overline{W}_1, \overline{W}_2) = (0.4, 0.325)$, we know the optimal average service cost is 13.5. Table 5.2 presents results under the DynRate policy for different values of parameter V; sample standard deviations are in parentheses. We observe that average service cost as well as the resulting mean queueing delay in each job class approaches optimal with the increase of V.

V	$\overline{W}_1$	$\overline{W}_2$	$\overline{P}$
1	0.3562 (0.00078)	0.3029 (0.00032)	13.8018 (0.01806)
10	0.3984 (0.00022)	0.3247 (0.00005)	13.5101 (0.02626)
100	0.4003 (0.00013)	0.3252 (0.00010)	13.5044 (0.02197)
Optimal:	0.4	0.325	13.5

Table 5.2: Simulation for the DynRate policy under different values of V.

CostDelayFair Policy

For the CostDelayFair policy, we solve

$$\text{minimize:} \quad \frac{1}{2}(\overline{W}_1)^2 + 2(\overline{W}_2)^2 \tag{5.57}$$

$$\text{subject to:} \quad \overline{P} \leq 13.5. \tag{5.58}$$

The optimal policy must satisfy (5.58) with equality. In Fig. 5.2, the set of all achievable mean queueing delay vectors with $\overline{P} = 13.5$ forms a line segment AB that is parallel to both delay regions $\mathcal{W}(16)$ and $\mathcal{W}(25)$, and passes $(0.4, 0.325)$. This can be shown geometrically in Fig. 5.3 by observing that any randomized policy that achieves some point on AB must have the same convex combination of one point on $\mathcal{W}(25)$ and one on $\mathcal{W}(16)$, and therefore incurs the same average service cost $\overline{P} = 13.5$.

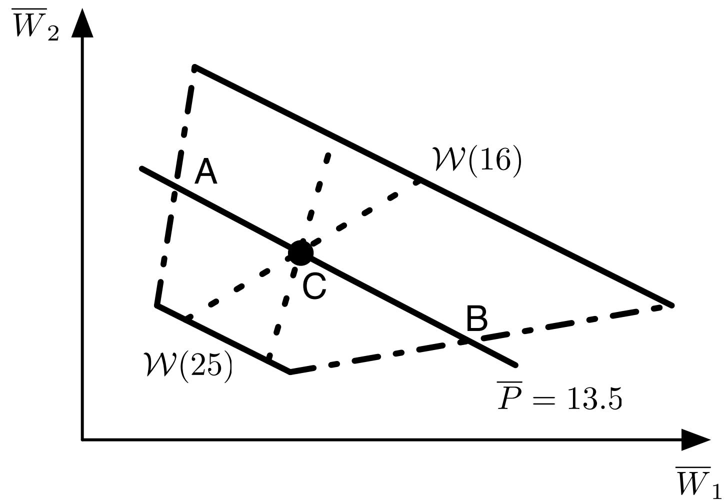

Figure 5.3: The two dotted lines passing point C on AB represent two arbitrary randomized policies that achieve C. Geometrically, they have the same mixture of one point on $\mathcal{W}(25)$ and one on $\mathcal{W}(16)$. Therefore, they incur the same average service cost.

Consequently, (5.57)-(5.58) is equivalent to

$$\text{minimize: } 0.5(\overline{W}_1)^2 + 2(\overline{W}_2)^2 \tag{5.59}$$

$$\text{subject to: } \overline{W}_1 + \overline{W}_2 = 1.05 \tag{5.60}$$

$$\overline{W}_1 \geq \frac{2}{15},\ \overline{W}_2 \geq \frac{23}{120}, \tag{5.61}$$

where the constraints (5.60)-(5.61) represent the line segment AB in Fig. 5.2. The optimal average delay vector that solves (5.59)-(5.61) is $(\overline{W}_1^*, \overline{W}_2^*) = (0.525, 0.2625)$. Table 5.3 presents the simulation results with sample standard deviations in parentheses. Again, the performance approaches optimal as V increases.

V	$\overline{W}_1$	$\overline{W}_2$	$\frac{1}{2}(\overline{W}_1)^2 + 2(\overline{W}_2)^2$	$\overline{P}$
100	0.5657 (0.0031)	0.3037 (0.0013)	0.3445 (0.0032)	13.0818 (0.0030)
200	0.5421 (0.0017)	0.2855 (0.0009)	0.3100 (0.0017)	13.2738 (0.0029)
500	0.5245 (0.0023)	0.2705 (0.0011)	0.2839 (0.0022)	13.4543 (0.0014)
1000	0.5195 (0.0022)	0.2651 (0.0010)	0.2755 (0.0022)	13.4963 (0.0006)
Optimal:	0.525	0.2625	0.2756	13.5

Table 5.3: Simulation for the CostDelayFair policy under different values of V.

5.8 Chapter Summary and Discussions

A multi-class queueing system is a useful analytical model to study how to share a common resource to different types of service requests to optimize a joint objective or provide differentiated services. Motivated by modern computer and cloud computing applications, we consider a nonpreemptive multi-class $M/G/1$ queue with adjustable service rates, and study several convex delay penalty and average service cost minimization problems with time-average constraints. After characterizing the performance region of queueing delay vectors and average service cost in the $M/G/1$ queue as a convex hull of performance vectors of a collection of frame-based policies, we use Lyapunov drift theory to construct dynamic $c\mu$ rules that have near-optimal performance. The time-varying parameters in the dynamic $c\mu$ rules capture the running delay performance in each job class. Intuitively, our approach turns objective functions and time average constraints into virtual queues that need to be stabilized. To develop queue-stable policies, we use the ratio MaxWeight method developed in Chapter 4 in every busy period. Our policies require limited statistical knowledge, and may be viewed as learning algorithms over queueing systems. These learning algorithms have a tradeoff between performance and learning time, controlled by a parameter $V > 0$.

Lyapunov drift theory is a new stochastic optimization toolset that has been developed over the years in the context of wireless networks and packet switches. We suspect that this toolset shall have wide applications in queueing systems, especially those satisfying conservation laws and having polymatroidal (or more generally, convex) performance regions. Different metrics such as throughput (together with admission control), delay, power, and functions of them may be mixed to serve as objective functions or time average constraints. It is of our interest to explore these directions.

In this chapter, we assume that job sizes are unknown upon their arrival. If job sizes are known in advance, the analysis developed in this chapter applies as well. The key is to define *subclasses* in each class so that every subclass consists of jobs having the same actual known service time, and we schedule job services across all subclasses. An important consequence is that, when job size information is available, the $c\mu$ rule over the original parent classes is no longer optimal for linear optimization problems such as minimizing a weighted sum of average queue sizes. In addition, in each original class, FIFO is no longer optimal; instead, we shall serve jobs using shortest job size first.

5.9 Bibliographic Notes

The analysis and control of queueing systems dates back to 1950s; see Niño-Mora [2009], Stidham [2002] for detailed surveys. The use of the achievable region approach for queue control, in contrast to the traditional use of Markov decision theory [Walrand 1988], starts in 1980s; see Bertsimas and Niño-Mora [1996] for a survey. In the achievable region method, many results consider unconstrained linear optimization problems, such as minimizing a weighted sum of average queue sizes across job classes. They generalize the idea that if the performance region satisfies conservation laws, which result in different notions of polymatroidal regions, linear problems are solved by simple index policies such as the $c\mu$ rule or an adaptive greedy policy. Unfortunately, these results seem difficult to apply to nonlinear optimization problems, even to linear ones with side constraints. Our analysis in this chapter takes advantage of the strong conservation law [Shanthikumar and Yao 1992] and the $c\mu$ rule in a nonpreemptive multi-class $M/G/1$ queue [Federgruen and Groenevelt 1988b], and uses them to develop dynamic $c\mu$ rules that solve convex delay minimization and optimal rate control problems with time-average constraints. We note that the strong conservation law does *not* hold for all time for the rate allocation problems we study. However, it holds locally over distinct busy periods, which we show is sufficient for our purposes.

The control of service rates in single-server single-class queues is previously studied by Ata and Shneorson [2006], George and Harrison [2001], Mitchell [1973], Rosberg et al. [1982], Stidham and Weber [1989], Weber and Stidham [1987]. The optimal allocation of state-dependent service rates is analyzed by Markov decision processes (MDP), and is shown to have monotone structures; namely, the service rate increases with the queue size. We note that applying state-dependent service rate control and MDP analysis in our multi-class queue control problems seems difficult. It involves a multi-dimensional state space and hence would suffer from the curse of dimensionality. We avoid this complexity explosion by exploiting the strong conservation law for unfinished work, which gives rise to our dynamic $c\mu$ policies that only update controls across busy periods. Furthermore, MDP analysis seems complicated to incorporate time-average delay and service cost constraints. It also requires full statistical knowledge of the queues, where our policies require no or limited statistics.

In prior work on convex optimization over queues, minimizing convex holding costs is treated as a restless bandit problem by Ansell et al. [2003] and Glazebrook et al. [2003], in which Whittle's index heuristics are constructed. A generalized $c\mu$ rule is shown asymptotically optimal for convex holding costs under heavy traffic by Gurvich and Whitt [2009], Mandelbaum and Stolyar [2004], van Mieghem [1995]. For the minimization of a separable convex function of the mean delay vector, Federgruen and Groenevelt [1988a] provides two numerical methods, but it is unclear how to control the system to achieve the optimal solution. Federgruen and Groenevelt [1988b] provides a synthesis algorithm to achieve a given feasible mean delay vector in a multi-class $M/G/c$ queue; this algorithm is complicated and requires second-order statistics of the queue. Bhattacharya et al. [1995] introduces a dynamic priority policy in which priorities are updated over intervals using the running performance in all job classes. This policy, based on stochastic approximation, has design philosophy similar to ours. But our policies are conceptually simpler and incorporate time average constraints and adjustable service rates.

Power-aware scheduling of computers is widely studied in different contexts in the literature, where two main analytical tools are competitive analysis [Andrew et al. 2010] and $M/G/1$-type queueing theory (see Wierman et al. [2009] and references therein), both used to optimize popular metrics such as a weighted sum of average power and average delay. This chapter presents a fundamentally different approach to study these problems. Our policies provide more directed control over average power cost and average delay, and consider a multi-class setup with time average constraints.

5.10 Additional Results in Chapter 5

5.10.1 Proof of Lemma 5.4

We index all $N!$ nonpreemptive strict priority policies by $\{\pi_j\}_{j\in\mathcal{J}}$, $\mathcal{J} = \{1, \dots, N!\}$. Consider a stationary randomized priority policy π_{rand}, defined by a probability distribution $\{\alpha_j\}_{j\in\mathcal{J}}$, that randomly chooses the jth priority policy π_j with probability α_j in every frame. Let $W_n^{\text{sum}}(\pi_j)$ denote the sum of queueing delays in class n during a frame in which policy π_j is used; we define $W_n^{\text{sum}}(\pi_{\text{rand}})$ similarly for policy π_{rand}. By conditional expectation, we get $\mathbb{E}\left[W_n^{\text{sum}}(\pi_{\text{rand}})\right] = \sum_{j\in\mathcal{J}} \alpha_j\, \mathbb{E}\left[W_n^{\text{sum}}(\pi_j)\right]$. Next, define $\overline{W}_n(\pi_j)$ as the average queueing delay for class n if policy π_j is used in every frame; define $\overline{W}_n(\pi_{\text{rand}})$ similarly under π_{rand}. It follows that

$$\overline{W}_n(\pi_{\text{rand}}) \overset{(a)}{=} \frac{\mathbb{E}\left[W_n^{\text{sum}}(\pi_{\text{rand}})\right]}{\lambda_n \mathbb{E}\left[T\right]} = \sum_{j\in\mathcal{J}} \alpha_j \frac{\mathbb{E}\left[W_n^{\text{sum}}(\pi_j)\right]}{\lambda_n \mathbb{E}\left[T\right]} \overset{(b)}{=} \sum_{j\in\mathcal{J}} \alpha_j\, \overline{W}_n(\pi_j). \tag{5.62}$$

where (a) and (b) follow renewal reward theory (cf. (5.5)), and $\mathbb{E}\left[T\right]$ denotes the mean frame size, independent of scheduling policies when the service rate is constant. Define $x_n(\pi_j) \triangleq \rho_n \overline{W}_n(\pi_j)$ for all strict priority policies π_j, and define $x_n(\pi_{\text{rand}})$ similarly. Multiplying (5.62) by ρ_n for all classes n yields $(x_n(\pi_{\text{rand}}))_{n=1}^N = \sum_{j\in\mathcal{J}} \alpha_j\, (x_n(\pi_j))_{n=1}^N$, which shows that the performance vector under policy π_{rand} is a convex combination of those under strict priority policies $\{\pi_j\}_{j\in\mathcal{J}}$. It proves the first part of the lemma.

Conversely, for any given vector $(\overline{W}_n)_{n=1}^N$ in the delay region $\mathcal{W}$, there exists a probability distribution $\{\beta_j\}_{j\in\mathcal{J}}$ such that $\rho_n \overline{W}_n = \sum_{j\in\mathcal{J}} \beta_j\, x_n(\pi_j)$ for all classes n, i.e., $\overline{W}_n = \sum_{j\in\mathcal{J}} \beta_j\, \overline{W}_n(\pi_j)$ for all n. From (5.62), the stationary randomized policy

π_{rand} defined by the probability distribution $\{\beta_j\}_{j\in\mathcal{J}}$ achieves the average delay vector $(\overline{W}_n)_{n=1}^N$. $\square$

5.10.2 Lemma 5.19

Lemma 5.19. In a nonpreemptive N-class $M/G/1$ queue with a constant service rate, if the first four moments of service times X_n are finite for all classes $n \in \{1, \ldots, N\}$, and the system is stable with $\sum_{n=1}^N \lambda_n \mathbb{E}[X_n] < 1$, then, in every frame $k \in \mathbb{Z}^+$, the term $\mathbb{E}\left[\left(\sum_{i\in A_{n,k}} \left(W_{n,k}^{(i)} - d_n\right)\right)^2\right]$ is finite for all classes n under any work-conserving policy.

Proof (Lemma 5.19)**.** For brevity, we only give a sketch of proof. Using $\mathbb{E}\left[(a-b)^2\right] \leq 2\mathbb{E}\left[a^2+b^2\right]$, it suffices to show that $\mathbb{E}\left[\left(\sum_{i\in A_{n,k}} W_{n,k}^{(i)}\right)^2\right]$ and $\mathbb{E}\left[|A_{n,k}|^2\right]$ are both finite. We only show the first expectation is finite; the finiteness of the second expectation then follows. Define $N_k \triangleq \sum_{n=1}^N |A_{n,k}|$ as the number of jobs (over all classes) served in frame k; we have $|A_{n,k}| \leq N_k$ for all n and k. In frame k, the queueing delay $W_{n,k}^{(i)}$ of a class n job $i \in A_{n,k}$ is less than or equal to the busy period size B_k. Then we get $\mathbb{E}\left[\left(\sum_{i\in A_{n,k}} W_{n,k}^{(i)}\right)^2\right] \leq \mathbb{E}\left[B_k^2 N_k^2\right]$. By Cauchy-Schwarz inequality, we have $\mathbb{E}\left[B_k^2 N_k^2\right] \leq \sqrt{\mathbb{E}\left[B_k^4\right]\mathbb{E}\left[N_k^4\right]}$. It suffices to show that both $\mathbb{E}\left[B_k^4\right]$ and $\mathbb{E}\left[N_k^4\right]$ are finite.

First we argue $\mathbb{E}\left[B_k^4\right] < \infty$. In the following we drop the index k for notational convenience. Since the busy period size B is the same under any work-conserving policy, it is convenient to consider LIFO scheduling with preemptive priority, and that jobs of all classes are treated equally likely. In this scheme, let a_0 denote the arrival that starts the current busy period. Arrival a_0 can be of any class, and the duration it stays in the system is equal to the busy period B. Next, let $\{a_1, \ldots, a_M\}$ denote the M jobs that arrive *during the service of job* a_0. Let $B(1), \ldots, B(M)$ denote the duration they stay in the system. Under LIFO with preemptive priority, we observe that $B(1), \ldots, B(M)$ are independent and identically distributed as the starting busy period B, since any new arrival *never sees* any previous arrivals and starts a new busy period (by the memoryless property of Poisson arrivals). Consequently, we have

$$B = X + \sum_{m=1}^{M} B(m), \tag{5.63}$$

where X denote the service time of a_0. Notice also that each duration $B(m)$ for all $m \in \{1, \ldots, M\}$ is independent of M. By taking the square and expectation of (5.63), we can compute $\mathbb{E}\left[B^2\right]$ in closed form and show that it is finite if the first two moments of service times X_n are finite for all n. Likewise, by raising (5.63) to the third and fourth power and taking expectation, we can compute $\mathbb{E}\left[B^3\right]$ and $\mathbb{E}\left[B^4\right]$ and show they are finite if the first four moments of X_n are finite (showing $\mathbb{E}\left[B^4\right] < \infty$ requires the finiteness of the first three moments of B).

Likewise, to show $\mathbb{E}\left[N^4\right]$ is finite, under LIFO with preemptive priority we observe

$$N = 1 + \sum_{m=1}^{M} N(m), \tag{5.64}$$

where $N(m)$ denotes the number of arrivals, including a_m, served during the stay of arrival a_m in the system; $N(m)$ are i.i.d. and independent of M. By raising (5.64) to the second, third, and fourth power and taking expectation, we can compute $\mathbb{E}\left[N^4\right]$ in closed form and show it is finite. $\square$

5.10.3 Proof of Lemma 5.14

From (5.20), we get

$$Y_{n,k+1} \geq Y_{n,k} - r_{n,k} \left|A_{n,k}\right| + \sum_{i \in A_{n,k}} W_{n,k}^{(i)}.$$

Summing over $k \in \{0, \ldots, K-1\}$ and using $Y_{n,0} = 0$ yield

$$\sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)} - Y_{n,K} \leq \sum_{k=0}^{K-1} r_{n,k} \left| A_{n,k} \right|.$$

Taking expectation and dividing by $\lambda_n \mathbb{E}\left[\sum_{k=0}^{K-1} T_k\right]$ yield

$$\frac{\mathbb{E}\left[\sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)}\right]}{\lambda_n \mathbb{E}\left[\sum_{k=0}^{K-1} T_k\right]} - \frac{\mathbb{E}\left[Y_{n,K}\right]}{\lambda_n K \mathbb{E}\left[T_0\right]} \leq \frac{\mathbb{E}\left[\sum_{k=0}^{K-1} r_{n,k} \left| A_{n,k} \right|\right]}{\lambda_n \mathbb{E}\left[\sum_{k=0}^{K-1} T_k\right]}. \tag{5.65}$$

where in the second term we have used $\mathbb{E}[T_k] = \mathbb{E}[T_0]$ for all k. In the last term of (5.65), since the value $r_{n,k}$ is chosen independent of $|A_{n,k}|$ and T_k, we use $\mathbb{E}[|A_{n,k}|] = \lambda_n \mathbb{E}[T_k]$ and get

$$\frac{\mathbb{E}\left[\sum_{k=0}^{K-1} r_{n,k} \left| A_{n,k} \right|\right]}{\lambda_n \mathbb{E}\left[\sum_{k=0}^{K-1} T_k\right]} = \frac{\mathbb{E}\left[\sum_{k=0}^{K-1} r_{n,k} T_k\right]}{\mathbb{E}\left[\sum_{k=0}^{K-1} T_k\right]}.$$

Define $\theta_K^{(n)}$ as the left side of (5.65). It follows

$$\theta_K^{(n)} \leq \frac{\mathbb{E}\left[\sum_{k=0}^{K-1} r_{n,k} T_k\right]}{\mathbb{E}\left[\sum_{k=0}^{K-1} T_k\right]}.$$

Since $f_n(\cdot)$ is nondecreasing for all classes n, we get

$$\limsup_{K \to \infty} \sum_{n=1}^{N} f_n\left(\theta_K^{(n)}\right) \leq \limsup_{K \to \infty} \sum_{n=1}^{N} f_n\left(\frac{\mathbb{E}\left[\sum_{k=0}^{K-1} r_{n,k} T_k\right]}{\mathbb{E}\left[\sum_{k=0}^{K-1} T_k\right]}\right). \tag{5.66}$$

Using (5.65), define the value

$$\eta_K^{(n)} \triangleq \frac{\mathbb{E}\left[\sum_{k=0}^{K-1} \sum_{i \in A_{n,k}} W_{n,k}^{(i)}\right]}{\mathbb{E}\left[\sum_{k=0}^{K-1} \left| A_{n,k} \right|\right]} = \theta_K^{(n)} + \frac{\mathbb{E}\left[Y_{n,K}\right]}{\lambda_n K \mathbb{E}\left[T_0\right]}. \tag{5.67}$$

To complete the proof, using (5.66) it suffices to show

$$\limsup_{K\to\infty} \sum_{n=1}^{N} f_n(\eta_K^{(n)}) = \limsup_{K\to\infty} \sum_{n=1}^{N} f_n(\theta_K^{(n)}). \tag{5.68}$$

We show that inequality $\leq$ holds in (5.68); the other direction is proved similarly. Let the left side of (5.68) attains its lim sup in the subsequence $\{K_m\}_{m=1}^{\infty}$. It follows

$$\begin{aligned}\limsup_{K\to\infty} \sum_{n=1}^{N} f_n(\eta_K^{(n)}) = \lim_{m\to\infty} \sum_{n=1}^{N} f_n(\eta_{K_m}^{(n)}) &\overset{(a)}{=} \sum_{n=1}^{N} f_n\left(\lim_{m\to\infty} \eta_{K_m}^{(n)}\right) \\ &\overset{(b)}{=} \sum_{n=1}^{N} f_n\left(\lim_{m\to\infty} \theta_{K_m}^{(n)}\right) \leq \limsup_{K\to\infty} \sum_{n=1}^{N} f_n(\theta_K^{(n)}),\end{aligned}$$

where (a) follows the continuity of $f_n(\cdot)$ for all classes n, and (b) follows (5.67) and mean rate stability of the $Y_{n,k}$ queues. □

5.10.4 Independence of Second-Order Statistics in DynRate

We show how to remove the dependence on the second moments of job sizes S_n in the DynRate policy in Section 5.5.1. For simplicity, we assume that job classes are properly re-ordered so that class n has the nth highest priority for $n \in \{1, \dots, N\}$. We rewrite (5.34) using (5.8) as

$$\hat{R}\left[\left(\frac{V}{\hat{R}} \sum_{n=1}^{N} \lambda_n \mathbb{E}\left[S_n\right]\right) \frac{P_k}{\mu(P_k)} + \sum_{n=1}^{N} \frac{Z_{n,k}\,\lambda_n}{(\mu(P_k) - \sum_{m=0}^{n-1} \hat{\rho}_m)(\mu(P_k) - \sum_{m=0}^{n} \hat{\rho}_m)}\right] \tag{5.69}$$

where

$$\hat{R} \triangleq \frac{1}{2} \sum_{n=1}^{N} \lambda_n \mathbb{E}\left[S_n^2\right], \ \hat{\rho}_m \triangleq \begin{cases} \lambda_m \mathbb{E}\left[S_m\right], & 1 \leq m \leq N \\ 0, & m = 0 \end{cases}$$

By ignoring constant $\hat{R}$ and redefining $\widetilde{V} \triangleq V/\hat{R}$ in (5.69), in the kth frame of policy DynRate, it is equivalent to allocate service rate $\mu(P_k)$ that minimizes

$$\left(\widetilde{V}\sum_{n=1}^{N}\lambda_n\mathbb{E}\left[S_n\right]\right)\frac{P_k}{\mu(P_k)}+\sum_{n=1}^{N}\frac{Z_{n,k}\,\lambda_n}{(\mu(P_k)-\sum_{m=0}^{n-1}\hat{\rho}_m)(\mu(P_k)-\sum_{m=0}^{n}\hat{\rho}_m)}, \tag{5.70}$$

The sum (5.70) is independent of second moments of job sizes. From Theorem 5.16 and using $V=\widetilde{V}\hat{R}$, this alternative policy yields average service cost $\overline{P}$ satisfying

$$\overline{P}\leq\frac{C\sum_{n=1}^{N}\lambda_n}{\widetilde{V}\hat{R}}+P^*,$$

and we preserve the property that the average service cost $\overline{P}$ is $O(1/\widetilde{V})$ away from the optimal P^*. □

5.10.5 Proof of Lemma 5.17

From (5.44), we have $X_{k+1}\geq X_k+P_kB_k(P_k)-P_{\text{const}}\,T_k(P_k)$. Summing over $k\in\{0,\ldots,K-1\}$, taking expectation, and using $X_0=0$, we get

$$\mathbb{E}\left[X_K\right]\geq\mathbb{E}\left[\sum_{k=0}^{K-1}P_kB_k(P_k)\right]-P_{\text{const}}\,\mathbb{E}\left[\sum_{k=0}^{K-1}T_k(P_k)\right].$$

Dividing by $\mathbb{E}\left[\sum_{k=0}^{K-1}T_k(P_k)\right]$ and passing $K\to\infty$ yields

$$\overline{P}\leq P_{\text{const}}+\limsup_{K\to\infty}\frac{\mathbb{E}\left[X_K\right]}{K}\frac{K}{\mathbb{E}\left[\sum_{k=0}^{K-1}T_k(P_k)\right]}\overset{(a)}{\leq}P_{\text{const}}+\limsup_{K\to\infty}\frac{\mathbb{E}\left[X_K\right]}{K}\sum_{n=1}^{N}\lambda_n\overset{(b)}{=}P_{\text{const}}$$

where (a) uses $\mathbb{E}\left[T_k(P_k)\right]\geq\mathbb{E}\left[I_k\right]=1/(\sum_{n=1}^{N}\lambda_n)$, and (b) follows mean rate stability of the virtual cost queue X_k. □

5.10.6 Proof of Theorem 5.18

Let π^* be the stationary randomized policy of rate allocation and priority assignment that solves (5.42)-(5.43). Let $(\overline{W}_n^*)_{n=1}^N$ be the optimal mean delay vector, and $\overline{P}^*$ the associated average service cost; we have $\overline{P}^* \leq P_{\text{const}}$. In frame k, the ratio $\frac{\Psi(\chi_k)}{\mathbb{E}[T_k(P_k)|\chi_k]}$ under policy π^* and the genie decisions $r_{n,k}^* = \overline{W}_n^*$ for all n and k is equal to

$$\sum_{n=1}^N Y_{n,k}\, \lambda_n \overline{W}_n^* + X_k \overline{P}^* - X_k P_{\text{const}} + \sum_{n=1}^N \Big(V f_n(\overline{W}_n^*) - Y_{n,k}\, \lambda_n \overline{W}_n^* \Big) \leq V \sum_{n=1}^N f_n(\overline{W}_n^*). \tag{5.71}$$

Since the $\mathsf{CostDelayFair}$ policy minimizes $\frac{\Psi(\chi_k)}{\mathbb{E}[T_k(P_k)|\chi_k]}$ in every frame k, this ratio under the $\mathsf{CostDelayFair}$ policy satisfies

$$\frac{\Psi(\chi_k)}{\mathbb{E}\left[T_k(P_k) \mid \chi_k\right]} \leq V \sum_{n=1}^N f_n(\overline{W}_n^*).$$

Then (5.46) under the $\mathsf{CostDelayFair}$ policy satisfies

$$\Delta(\chi_k) + V\mathbb{E}\left[\sum_{n=1}^N f_n(r_{n,k})\, T_k(P_k) \mid \chi_k\right] \leq C + V\mathbb{E}\left[T_k(P_k) \mid \chi_k\right] \sum_{n=1}^N f_n(\overline{W}_n^*). \tag{5.72}$$

Removing the second term in (5.72) and taking expectation, we get

$$\mathbb{E}\left[L(\chi_{k+1})\right] - \mathbb{E}\left[L(\chi_k)\right] \leq C + V\mathbb{E}\left[T_k(P_k)\right] \sum_{n=1}^N f_n(\overline{W}_n^*).$$

Summing over $k \in \{0, \dots, K-1\}$, and using $L(\chi_0) = 0$ yields

$$\mathbb{E}\left[L(\chi_K)\right] \leq KC + V\mathbb{E}\left[\sum_{k=0}^{K-1} T_k(P_k)\right] \sum_{n=1}^N f_n(\overline{W}_n^*) \leq KC_1 \tag{5.73}$$

where $C_1 \triangleq C + V\mathbb{E}\left[T_0(P_{\min})\right] \sum_{n=1}^N f_n(\overline{W}_n^*)$, and we have used $\mathbb{E}\left[T_k(P_k)\right] \leq \mathbb{E}\left[T_0(P_{\min})\right]$ for all k. Inequality (5.73) suffices to conclude that the virtual cost queue X_k and all

$Y_{n,k}$ queues are mean rate stable. From Lemma 5.17 we have $\overline{P} \leq P_{\text{const}}$. The proof of (5.48) follows that of Theorem 5.12. $\square$

Chapter 6

Conclusions

This dissertation addresses single-hop network control problems in wireless networks and queueing systems. In wireless, we optimize channel probing considering its energy and timing overhead, and build dynamic strategies that are both throughput and energy optimal. We explore efficient exploitation of channel memory over Markov ON/OFF channels with unknown current states by studying the associated network capacity region. Computing the full region is difficult, and we construct a good inner capacity bound with easily achievable closed-form performance and asymptotical optimality in special cases. Queue-dependent throughput-optimal policies are provided within the inner bound. We also solve throughput utility maximization over the inner bound, because of its wide applications, with the use of a novel ratio MaxWeight policy. In queueing systems, motivated by computer and cloud computing applications, we solve several convex delay penalty and service cost minimization problems with time-average constraints in a multi-class $M/G/1$ queue with adjustable service rates. We build dynamic priority policies that greedily update controls over busy periods based on running delay performance, and prove that they are near-optimal.

We use and develop a dynamic optimization method throughout. First we characterize the performance region. We explore the problem structure to identify a policy space, as simple as we could possibly get, so that each feasible performance vector is supported by a stationary randomization over it. For this step, sample-path analysis and the strong conversation law in multi-class queues are used in Chapter 2 and Chapter 5, respectively. When the performance region is hard to compute, we use the problem structure to *design* a policy space, and settle for a good achievable region by randomizing policies in that space; this is what we have done in Chapter 3 and Chapter 4. Overall, a simple *policy-based* representation of a performance region facilitates us to develop good control policies that scale with the dimension of the problem.

Having a policy space corresponding to the performance region, we use Lyapunov drift theory to construct provably optimal network control algorithms. At every decision epoch, one policy in that policy space is greedily chosen for use until the next decision epoch, based on the running system performance summarized in the queueing information. The definition of decision epochs depends on how the policy space is structured, and the duration between two consecutive decision epochs may cross one slot, one busy period, or a policy-dependent random frame size. The resulting control algorithms require limited or no a-priori statistical knowledge of the system, implying that they adapt nicely to unknown system parameters, and can be treated as learning algorithms over stochastic queueing networks.

Perhaps the most important lesson we have learned is the ability to use the analytical framework we have generalized to solve open stochastic convex optimization problems with time-average constraints. Chapter 5 is committed to such an attempt with application to the control of multi-class queues. We transform convex objective functions and time-average constraints into virtual queues that need to be stabilized. Guided by the dynamics in the virtual queues, we construct queue-stable policies that solve the original problems.

Bibliography

Ahmad, S. H. A. and Liu, M. (2009). Multi-channel opportunistic access: A case of restless bandits with multiple plays. In *Allerton Conf. Communication, Control, and Computing*, pages 1361–1368. 83

Ahmad, S. H. A., Liu, M., Javidi, T., Zhao, Q., and Krishnamachari, B. (2009). Optimality of myopic sensing in multichannel opportunistic access. *IEEE Trans. Inf. Theory*, 55(9):4040–4050. 7, 8, 56, 57, 62, 73, 83, 84

Andrew, L. L. H., Lin, M., and Wierman, A. (2010). Optimality, fairness, and robustness in speed scaling designs. In *ACM SIGMETRICS*. 166

Ansell, P. S., Glazebrook, K. D., Mitrani, I., and Nino-Mora, J. (1999). A semidefinite programming approach to the optimal control of a single server queueing system with imposed second moment constraints. *Journal of the Oper. Res.*, 50(7):765–773. 128

Ansell, P. S., Glazebrook, K. D., Niño-Mora, J., and O'Keeffe, M. (2003). Whittle's index policy for a multi-class queueing system with convex holding costs. *Math. Methods of Oper. Res.*, 57:21–39. 166

Ata, B. and Shneorson, S. (2006). Dynamic control of an M/M/1 service system with adjustable arrival and service rates. *Manage. Sci.*, 52(11):1778–1791. 165

Bertsekas, D. P. (1999). *Nonlinear Programming*. Athena Scientific, 2nd edition. 140

Bertsekas, D. P. (2005). *Dynamic Programming and Optimal Control*, volume I. Athena Scientific, 3rd edition. 7, 54, 118

Bertsekas, D. P. and Gallager, R. G. (1992). *Data Networks*. Prentice Hall, 2nd edition. 133, 134, 158

Bertsimas, D. and Niño-Mora, J. (1996). Conservation laws, extended polymatroids, and multi-armed bandit problems; a polyhedral approach to indexable systems. *Math. of Oper. Res.*, 21(2):257–306. 165

Bhattacharya, P. P., Georgiadis, L., and Tsoucas, P. (1995). Problems of adaptive optimization in multiclass M/GI/1 queues with bernoulli feedback. *Math. of Oper. Res.*, 20(2):355–380. 166

Buzacott, J. A. and Shanthikumar, J. G. (1993). *Stochastic Models of Manufacturing Systems*. Prentice Hall. 4

Chang, N. B. and Liu, M. (2007). Optimal channel probing and transmission scheduling for opportunistic spectrum access. In *ACM Int. Conf. Mobile Computing and Networking (MobiCom)*, pages 27–38, New York, NY. 40, 119

Chaporkar, P. and Proutiere, A. (2008). Optimal joint probing and transmission strategy for maximizing throughput in wireless systems. *IEEE J. Sel. Areas Commun.*, 26(8):1546–1555. 119

Chaporkar, P., Proutiere, A., and Asnani, H. (2010). Learning to optimally exploit multi-channel diversity in wireless systems. In *IEEE Proc. INFOCOM*. 119

Chaporkar, P., Proutiere, A., Asnani, H., and Karandikar, A. (2009). Scheduling with limited information in wireless systems. In *ACM Int. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*, New Orleans, LA. 84, 119

Eryilmaz, A. and Srikant, R. (2006). Joint congestion control, routing, and mac for stability and fairness in wireless networks. *IEEE J. Sel. Areas Commun.*, 24(8):1514–1524. 9, 119

Eryilmaz, A. and Srikant, R. (2007). Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. *IEEE/ACM Trans. Netw.*, 15(6):1333–1344. 9, 119

Federgruen, A. and Groenevelt, H. (1988a). Characterization and optimization of achievable performance in general queueing systems. *Oper. Res.*, 36(5):733–741. 166

Federgruen, A. and Groenevelt, H. (1988b). M/G/c queueing systems with multiple customer classes: Characterization and control of achievable performance under nonpreemptive priority rules. *Manage. Sci.*, 34(9):1121–1138. 9, 133, 134, 165, 166

Gallager, R. G. (1996). *Discrete Stochastic Processes*. Kluwer Academic Publishers. 55

Gelenbe, E. and Mitrani, I. (1980). *Analysis and Synthesis of Computer Systems*. Academic Press, London, UK. 9

George, J. M. and Harrison, J. M. (2001). Dynamic control of a queue with adjustable service rate. *Oper. Res.*, 49(5):720–731. 165

Georgiadis, L., Neely, M. J., and Tassiulas, L. (2006). Resource allocation and cross-layer control in wireless networks. *Foundations and Trends in Networking*, 1(1). 9, 47, 80

Gesbert, D. and Alouini, M.-S. (2004). How much feedback is multi-user diversity really worth? In *IEEE Int. Conf. Communications (ICC)*, pages 234–238. 40

Giaccone, P., Prabhakar, B., and Shah, D. (2003). Energy constrained wireless switching. In *Allerton Conf. Communication, Control, and Computing*, Urbana, IL. 39

Gittins, J. C. (1989). *Multi-Armed Bandit Allocation Indices*. Wiley, New York, NY. 71

Glazebrook, K. D., Lumley, R. R., and Ansell, P. S. (2003). Index heuristics for multiclass M/G/1 systems with nonpreemptive service and convex holding costs. *Queueing Syst.*, 45(2):81–111. 166

Gopalan, A., Caramanis, C., and Shakkottai, S. (2007). On wireless scheduling with partial channel-state information. In *Allerton Conf. Communication, Control, and Computing*. 40

Guha, S., Munagala, K., and Sarkar, S. (2006a). Approximation schemes for information acquisition and exploitation in multichannel wireless networks. In *Allerton Conf. Communication, Control, and Computing*. 40

Guha, S., Munagala, K., and Sarkar, S. (2006b). Jointly optimal transmission and probing strategies for multichannel wireless systems. In *Conf. Information Sciences and Systems*. 40, 119

Guha, S., Munagala, K., and Shi, P. (2009). Approximation algorithms for restless bandit problems. Technical report. 83, 84

Gurvich, I. and Whitt, W. (2009). Scheduling flexible servers with convex delay costs in many-server service systems. *Manufacturing Service Operations Management*, 11(2):237–253. 166

Jagannathan, K., Menache, I., Modiano, E., and Mannor, S. (2011). A state action frequency approach to throughput maximization over uncertain wireless channels. In *IEEE INFOCOM (Mini Conference)*, Shanghai, China. 84

Ji, Z., Yang, Y., Zhou, J., Takai, M., and Bagrodia, R. (2004). Exploiting medium acess diversity in rate adaptive wireless lans. In *ACM Int. Conf. Mobile Computing and Networking (MobiCom)*, pages 345–359, Philadelphia, PA. 39

Kar, K., Luo, X., and Sarkar, S. (2008). Throughput-optimal scheduling in multichannel access point networks under infrequent channel measurements. *IEEE Trans. Wireless Commun.*, 7(7):2619–2629. 40

Kaxiras, S. and Martonosi, M. (2008). *Computer Architecture Techniques for Power-Efficiency*. Synthesis Lectures on Computer Architecture. Morgan & Claypool. 6

Kelly, F. P. (1997). Charging and rate control for elastic traffic. *European Trans. Telecommunications*, 8:33–37. 6, 107, 141

Kleinrock, L. (1976). *Queueing Systems*, volume II: Computer Applications. Wiley Interscience. 4, 158

Lee, J.-W., Mazumdar, R. R., and Shroff, N. B. (2006). Opportunistic power scheduling for dynamic multi-server wireless systems. *IEEE Trans. Wireless Commun.*, 5(6):1506–1515. 39

Li, C.-P. and Neely, M. J. (2007). Energy-optimal scheduling with dynamic channel acquisition in wireless downlinks. In *IEEE Conf. Decision and Control (CDC)*, pages 1140–1147, New Orleans, LA. 12

Li, C.-P. and Neely, M. J. (2010a). Energy-optimal scheduling with dynamic channel acquisition in wireless downlinks. *IEEE Trans. Mobile Comput.*, 9(4):527 –539. 12, 53, 62, 73, 84, 119

Li, C.-P. and Neely, M. J. (2010b). Exploiting channel memory for multi-user wireless scheduling without channel measurement: Capacity regions and algorithms. In *IEEE Proc. Int. Symp. Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, Avignon, France. 12

Li, C.-P. and Neely, M. J. (2011a). Exploiting channel memory for multiuser wireless scheduling without channel measurement: Capacity regions and algorithms. *Performance Evaluation*. accepted for publication. 12, 125

Li, C.-P. and Neely, M. J. (2011b). Network utility maximization over partially observable Markovian channels. In *IEEE Proc. Int. Symp. Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, Princeton, NJ, USA. 9, 12

Liu, K. and Zhao, Q. (2010). Indexability of restless bandit problems and optimality of whittle's index for dynamic multichannel access. *IEEE Trans. Inf. Theory*, 56(11):5547–5567. 83, 84

Mandelbaum, A. and Stolyar, A. L. (2004). Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized $c\mu$-rule. *Oper. Res.*, 52(6):836–855. 166

Mitchell, B. (1973). Optimal service-rate selection in an M/G/$\hat{1}$ queue. *SIAM J. Applied Mathematics*, 24(1):19–35. 165

Neely, M. J. (2003). *Dynamic Power Allocation and Routing for Satellite and Wireless Networks with Time Varying Channels*. PhD thesis, Massachusetts Institute of Technology. 9, 119, 121

Neely, M. J. (2006). Energy optimal control for time varying wireless networks. *IEEE Trans. Inf. Theory*, 52(7):2915–2934. 9, 13, 14, 20, 39, 41

Neely, M. J. (2010a). Dynamic optimization and learning for renewal systems. In *Asilomar Conf. Signals, Systems, and Computers*. 130

Neely, M. J. (2010b). Stability and capacity regions for discrete time queueing networks. arxiv report. 126

Neely, M. J. (2010c). *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool. 9, 111, 147

Neely, M. J., Modiano, E., and Li, C.-P. (2008). Fairness and optimal stochastic control for heterogeneous networks. *IEEE/ACM Trans. Netw.*, 16(2):396–409. 9, 107, 108, 119, 120

Neely, M. J., Modiano, E., and Rohrs, C. E. (2003). Power allocation and routing in multibeam satellites with time-varying channels. *IEEE/ACM Trans. Netw.*, 11(1):138–152. 23, 39, 47

Neely, M. J., Modiano, E., and Rohrs, C. E. (2005). Dynamic power allocation and routing for time-varying wireless networks. *IEEE J. Sel. Areas Commun.*, 23(1):89–103. 2, 51

Neely, M. J. and Urgaonkar, R. (2008). Opportunism, backpressure, and stochastic optimization with the wireless broadcast advantage. In *Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA. 32

Niño-Mora, J. (2008). An index policy for dynamic fading-channel allocation to heterogeneous mobile users with partial observations. In *Next Generation Internet Networks*, pages 231–238. 83, 84

Niño-Mora, J. (2009). Stochastic scheduling. In Floudas, C. A. and Pardalos, P. M., editors, *Encyclopedia of Optimization*, pages 3818–3824. Springer, 2nd edition. 165

Ny, J. L., Dahleh, M., and Feron, E. (2008). Multi-uav dynamic routing with partial observations using restless bandit allocation indices. In *American Control Conference*, Seattle, WA, USA. 106

Pantelidou, A., Ephremides, A., and Tits, A. L. (2007). Joint scheduling and routing for ad-hoc networks under channel state uncertainty. In *IEEE Proc. Int. Symp. Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. 84

Patil, S. and de Veciana, G. (2007). Reducing feedback for opportunistic scheduling in wireless systems. *IEEE Trans. Wireless Commun.*, 6(12):4227–4232. 40

Rosberg, Z., Varaiya, P. P., and Walrand, J. C. (1982). Optimal control of service in tandem queues. *IEEE Trans. Autom. Control*, ac-27(3):600–610. 165

Ross, S. M. (1996). *Stochastic Processes*. Wiley, 2 edition. 132

Rudin, W. (1976). *Principles of Mathematical Analysis*. McGraw-Hill, 3rd edition. 42, 43

Sabharwal, A., Khoshnevis, A., and Knightly, E. (2007). Opportunistic spectral usage: Bounds and a multi-band CSMA-CA protocol. *IEEE/ACM Trans. Netw.*, 15(3):533–545. 39

Shanthikumar, J. G., Ding, S., and Zhang, M. T. (2007). Queueing theory for semiconductor manufacturing systems: A survey and open problems. *IEEE Trans. Autom. Sci. Eng.*, 4(4):513–522. 4

Shanthikumar, J. G. and Yao, D. D. (1992). Multiclass queueing systems: Polymatroidal structure and optimal scheduling control. *Oper. Res.*, 40(2):S293–S299. 165

Stidham, S. (2002). Analysis, design, and control of queueing systems. *Oper. Res.*, 50(1):197–216. 165

Stidham, S. and Weber, R. (1989). Monotonic and insensitive optimal policies for contorl of queues with undiscounted costs. *Oper. Res.*, 37(4):611–625. 165

Stolyar, A. L. (2005). Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Syst.*, 50(4):401–457. 9, 119

Tang, T. and Heath, Jr., R. W. (2005). Opportunistic feedback for downlink multiuser diversity. *IEEE Commun. Lett.*, 9(10):948–950. 40

Tassiulas, L. (1997). Scheduling and performance limits of networks with constantly changing topology. *IEEE Trans. Inf. Theory*, 43(3):1067–1073. 2, 51

Tassiulas, L. and Ephremides, A. (1992). Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Trans. Autom. Control*, 37(12):1936–1948. 9

Tassiulas, L. and Ephremides, A. (1993). Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Trans. Inf. Theory*, 39(2):466–478. 2, 9, 19, 41, 72

U.S. Environmental Protection Agency (2007). Report to congress on server and data center energy efficiency public law 109-431. 5

van Mieghem, J. A. (1995). Dynamic scheduling with convex delay costs: The generalized cmu rule. *Ann. Appl. Probab.*, 5(3):809–833. 166

Walrand, J. (1988). *An Introduction to Queueing Networks*. Prentice Hall. 165

Wang, H. S. and Chang, P.-C. (1996). On verifying the first-order markovian assumption for a rayleigh fading channel model. *IEEE Trans. Veh. Technol.*, 45(2):353–357. 51

Wang, W.-H., Palaniswami, M., and Low, S. H. (2006). Application-oriented flow control: Fundamentals, algorithms, and fairness. *IEEE/ACM Trans. Netw.*, 14(6):1282–1291. 6

Weber, R. and Stidham, S. (1987). Optimal control of service rates in networks of queues. *Adv. Appl. Probab.*, 19(1):202–218. 165

Welsh, D. J. A. (1976). *Matroid Theory*. Academic Press, London, UK. 9

Whittle, P. (1988). Restless bandits: Activity allocation in a changing world. *J. Appl. Probab.*, 25:287–298. 7, 83, 84, 118

Wierman, A., Andrew, L. L. H., and Tang, A. (2009). Power-aware speed scaling in processor sharing systems. In *IEEE Proc. INFOCOM*, pages 2007–2015, Rio de Janeiro, Brazil. 166

Yao, D. D. (2002). Dynamic scheduling via polymatroid optimization. In *Performance Evaluation of Complex Systems: Techniques and Tools, Performance 2002, Tutorial Lectures*, pages 89–113, London, UK. Springer-Verlag. 133

Yeh, E. M. and Cohen, A. S. (2004). Throughput optimal power and rate control for queued multiaccess and broadcast communications. In *IEEE Int. Symp. Information Theory (ISIT)*, page 112, Chicago, IL. 39

Ying, L. and Shakkottai, S. (2008). On throughput optimality with delayed network-state information. In *Information Theory and Application Workshop (ITA)*, pages 339–344. 84

Ying, L. and Shakkottai, S. (2009). Scheduling in mobile ad hoc networks with topology and channel-state uncertainty. In *IEEE Proc. INFOCOM*, Rio de Janeiro, Brazil. 84

Zhao, Q., Krishnamachari, B., and Liu, K. (2008). On myopic sensing for multi-channel opportunistic access: Structure, optimality, and preformance. *IEEE Trans. Wireless Commun.*, 7(12):5431–5440. 8, 62, 72, 83, 84

Zhao, Q. and Sadler, B. M. (2007). A survey of dynamic spectrum access. *IEEE Signal Process. Mag.*, 24(3):79–89. 106

Zhao, Q. and Swami, A. (2007). A decision-theoretic framework for opportunistic spectrum access. *IEEE Wireless Commun. Mag.*, 14(4):14–20. 83, 106

Zorzi, M., Rao, R. R., and Milstein, L. B. (1996). A Markov model for block errors on fading channels. In *IEEE Personal, Indoor and Mobile Radio Communications Symp. (PIMRC)*. 51